

ECOLE DOCTORALE DES MINES DE PARIS - PARISTECH
INRIA - SOPHIA-ANTIPOLIS

THÈSE

présentée en première version en vue d'obtenir le grade de Docteur en sciences,
spécialité « Informatique temps réel, robotique et automatique »

par

Julien Hubert

MANIPULATEURS PARALLÈLES, SINGULARITÉS ET ANALYSE STATIQUE

Thèse soutenue le 28 octobre 2010 devant le jury composé de :

M. MARCO CARRICATO	(Rapporteur)
M. NICOLAS ANDREFF	(Rapporteur)
M. PHILIPPE WENGER	(Examineur)
M. LOTFI ROMDHANE	(Examineur)
M. ADOLFO SUAREZ ROOS	(Examineur)

À ma famille et aux amis

REMERCIEMENTS

Je tiens tout d'abord à remercier le directeur de cette thèse, Jean-Pierre Merlet, pour m'avoir fait confiance malgré les connaissances plutôt légères que j'avais en septembre 2007 sur la robotique parallèle, puis pour m'avoir guidé, encouragé, conseillé, fait voyager pendant ces trois années tout en me laissant une grande liberté et en me transmettant une petite partie de sa grande érudition.

Mes remerciements vont également à David Daney, pour la gentillesse et la grande disponibilité qu'il a manifestées à mon égard durant cette thèse, pour tous ses conseils et les idées qu'il a bien voulu partager. Tout comme Jean-Pierre, je lui dois beaucoup et travailler avec ces deux personnes fut une grande expérience aussi bien sur le plan professionnel que sur le plan humain.

Je ne sais d'ailleurs comment véritablement exprimer ma gratitude envers eux, le contexte formel de ces remerciements mais aussi une certaine pudeur et un profond respect me poussent à la simplicité. Quoi de plus simple donc qu'un grand : "MERCI!"

Pour avoir accepté de participer au jury de soutenance je tiens à remercier M. Phillipe Wenger, M. Lotfi Romdhane, M. Adolfo Suarez Roos ainsi que M. Nicolas Andreff et M. Marco Carricato qui ont également accepté d'être les rapporteurs de cette thèse et ainsi contribué à l'amélioration de mon manuscrit par leurs nombreuses remarques.

Je tiens également à remercier le professeur Ilian Bonev pour m'avoir accueilli deux mois au sein de son équipe.

C'est également avec un grand plaisir que je tiens à remercier tous les autres membres de l'équipe COPRIN pour leur gentillesse, leur disponibilité et leur sens de l'humour qui fut largement appréciable lors des repas quotidiens. Une petite pensée particulière pour Gilles Trombettoni avec qui j'eus l'occasion de mettre les gants, Nicolas Chleq pour quelques sessions grimpes dans cette belle région, Christine Claux et Nathalie Woodward pour leur assistance administrative, Bertrand Neveu pour m'avoir donné l'occasion d'assister à un opéra, Odile Pourtallier pour ses délicieux abricots et Yves Papegay pour sa bonne humeur. Enfin comment ne pas souligner l'apport de mes différents collègues de bureaux qui, bien que non permanents, furent d'un grand secours et d'une grande patience à mon égard lors de ces délicieux jours de mauvaises humeur. Pour tous les moments d'échanges qui jalonnèrent mon quotidien mais également pour le piètre sens de l'humour que je partage avec certains (deux devraient se reconnaître) je leur adresse à tous un joyeux salut amical.

Pour ce qui concerne mon sale caractère, je tiens également à remercier mon éminent "cooloc" dont le bronzage iconoclaste et la chevelure dorée éclaboussent les eaux chlorées de la commune antiboise. Merci pour ta patience à mon égard, tu es sûrement celui qui a le plus supporté mes humeurs mais comme je pense avoir fait de même je nous considère quitte la dessus. Mais c'est surtout pour tes innombrables tentatives de corruptions réussies, pour ta façon de prononcer cancoyotte, pour cultiver le paradoxe au point d'être à la fois maniaque et ingérable, pour toutes ces anecdotes où j'ai côtoyé un véritable thaumaturge de l'absurde, que je te remercie profondément.

Un grand merci aussi à tous les potes, les amis, les gens croisés à l'INRIA ou non pendant ces trois années à qui j'adresse une dédicace particulière par ordre aléatoire de préférences : Seb, Vincent, GFG, Raph, Quentin, Abdou, GJ, Jacky, Mastaaz, Ben, Sylvain, Alys, Emilie, Jeannot, Rania, Mel, Arnaud des abeilles, Flo, Vivien, Andrea, Jean, Alex M., Clive, John, Bapt, Mirko, Barbie, Sapna, Marie, Nico S., Nico T., Dam's, Clem, Alex L., Sonia, Antoine, Guillaume, Mandar, Bogdan, Thibaud, Ados, Ophélie, Tutu et Claire, Grivox, Jox et enfin le Yox.

Enfin je tiens à remercier toute ma famille pour leur indéfectible soutien. La liste de leurs bonnes attentions étant trop longue on se contentera de cette liste non exhaustive. Commençons par les bons petits plats de ma Mômman suivis par le petit rhum échappé du coffre magique de mon Pôpa souvent lors des matchs de ligue des champions. Continuons avec les milles et un conseils cinématographiques de mon frangin préféré toujours en vadrouille et que je vois malheureusement trop peu à mon goût. On pourrait en dire aulant de ma tante Cricri dont la gentillesse peuple d'innombrables souvenirs de mon enfance. N'oublions pas mon cousin Thierry et sa joyeuse petite famille (Sandrine, Emilie et Chloé) pour la vie qu'ils apportent lors des réunions familiales. Comme il est d'usage de garder le meilleur pour la fin, je tiens à remercier cette doyenne à l'âge canonique, ce cordon bleu à l'appétit féroce qui possède un lever de coude dont je mesure encore mal l'amplitude. Un grand merci à cette joueuse de belote à l'étrange subtilité mais qui n'en demeure pas moins la meilleure des grand-mères.

TABLE DES MATIÈRES

Table des matières	viii
1 Introduction	1
Introduction	1
1.1 Introduction	1
1.2 Les robots parallèles	3
1.2.1 Définition :	3
1.2.2 Avantages et inconvénients des robots parallèles	3
1.3 Les singularités	5
1.4 Contributions	7
2 Notations, Prérequis et Singularités	9
2.1 Notations	9
2.1.1 Nomenclature d'un mécanisme	9
2.1.2 Notations	9
2.2 Modèles géométriques	11
2.2.1 Modèle géométrique inverse	11
2.2.2 Modèle géométrique direct	12
2.3 Modèles cinématiques	12
2.4 Singularités des robots parallèles	14
2.5 Géométrie de Grassmann	15
2.5.1 Vecteur de Plücker	16
2.5.2 Jacobienne cinématique inverse	16
2.5.3 Modèle statique	17
2.5.4 Singularités et géométrie de Grassmann	18
2.6 Transition	19
3 Espace de travail statique du 3-RPR	21
3.1 Introduction	21
3.2 Définition de l'espace de travail statique	22
3.2.1 Espace de travail statique du 3-RPR	24
3.3 Frontière de l'espace de travail statique	24

3.4	Eléments de la frontière	26
3.5	Calcul de la frontière	26
3.5.1	Les points clés	27
3.5.2	Détermination des points clés	28
3.5.3	Les points clés auxiliaires	28
3.5.4	Appartenance d'une pose à la frontière	29
3.5.5	Principe théorique du calcul de la frontière	30
3.5.6	Mise en pratique	30
3.5.7	Variation de l'algorithme	34
3.6	Résultats	35
3.6.1	Exemple d'espace de travail statique	35
3.6.2	Temps de calcul	37
3.7	Conclusion	38
4	Introduction à l'analyse par intervalles	41
4.1	Introduction	41
4.2	Définitions élémentaires	42
4.2.1	Théorie des ensembles et intervalles	44
4.2.2	Arithmétique par intervalles	45
4.2.2.1	Opérations usuelles	45
4.2.3	Fonctions et intervalles	47
4.3	Un enjeu primordial : l'évaluation	48
4.3.1	Les causes d'évaluation non optimale	48
4.3.1.1	L'effet enveloppant	48
4.3.1.2	Occurrences multiples de variables	49
4.3.2	Utilisation de la formule de Taylor dans le cas de l'analyse par intervalles	50
4.3.3	Utilisation de la monotonie	51
4.4	Résolution d'un système d'équation en utilisant l'analyse par intervalles .	52
4.4.1	Les différents types de boîtes	52
4.4.2	L'algorithme classique de résolution : Evaluation-Bisection	53
4.4.3	Méthodes de filtrage	54
4.4.3.1	La méthode de contraction 2B	54
4.4.3.2	La méthode de filtrage reposant sur le <i>rognage</i> ou <i>shaving</i>	55
4.5	Résolution d'un système linéaire carré d'équations	56
4.5.1	Préconditionnement	57
4.5.2	Résolution	57
4.5.2.1	Le pivot de Gauss	58
4.5.2.2	La méthode Gauss-Seidel	58
4.6	Transition	59
5	Approximation de l'espace de travail statique par l'analyse par intervalles	61
5.1	Introduction	61
5.2	Objectifs	61

5.3	Boîte et espace de travail statique	62
5.3.1	Test d'inclusion	62
5.3.2	Tests d'exclusion	63
5.3.2.1	Tests par résolution du système linéaire intervalle	63
5.3.2.2	Test par évaluation du torseur statique	63
5.3.2.3	Test de Rohn	64
5.4	Algorithme de calcul de l'espace de travail statique	64
5.4.1	Gestion de la bisection	65
5.4.2	Gestion du Stockage	66
5.4.3	Qualité de l'approximation de W_S	67
5.4.4	Aspect incrémental et amélioration de la qualité	67
5.4.5	Implantation distribuée	68
5.5	Éléments pour l'implantation de différentes stratégies de résolution	69
5.5.1	L'évaluation	69
5.5.1.1	Différentes stratégies pour l'évaluation	69
5.5.1.2	Formulation Matrice	70
5.6	Améliorer l'algorithme	70
5.6.1	Étape du point de choix	70
5.6.2	Contraintes supplémentaires	71
5.6.3	Stockage de données	71
5.6.3.1	Stockage des valeurs des τ_i	71
5.6.3.2	Stockage du signe des dérivées	71
5.6.3.3	Stockage des contraintes vérifiées	72
5.6.4	Améliorer les tests d'inclusions et d'exclusions impliquant la résolution du système linéaire	72
5.6.5	Ajout d'une méthode de type rognage	73
5.6.5.1	Le rognage IN-OUT	73
5.6.5.2	Propriétés dérivant du principe fondamental	73
5.6.5.3	Implantation d'un algorithme de rognage <i>IN-OUT</i>	75
5.6.6	Bisection multi-niveaux	76
5.6.6.1	La structure pour le multi-niveaux	76
5.6.6.2	Algorithme multi-niveaux	77
5.7	Adaptation de l'algorithme pour répondre à différents problèmes	78
5.7.1	Prise en compte d'incertitude sur les paramètres géométriques	78
5.7.2	Vérification de l'inclusion d'une zone dans W_s	79
5.8	Résultats	79
5.9	Conclusion	88
6	Conclusion et Perspectives	91
6.1	Conclusion	91
6.2	Perspectives	92
	Bibliographie	95

NOTATIONS

Robotique

- \times : produit en croix de deux vecteurs
- \cdot : produit scalaire de deux vecteurs
- χ : vecteur des coordonnées généralisées d'un robot
- Θ : coordonnées articulaires d'un robot
- Θ_a : coordonnées articulaires actives d'un robot
- Θ_p : coordonnées articulaires passives d'un robot
- ρ_i : usuellement la longueur de la i -ème jambe du robot
- ρ_{min}, ρ_{max} : longueur minimale et maximale des jambes du robots.
- Ω_O ($O, \mathbf{x}, \mathbf{y}, \mathbf{z}$) : repère de référence, généralement lié à la base du robot.
- Ω_C (C, x_r, y_r, z_r) : repère mobile du robot souvent lié à la plate-forme du robot.
- C : origine du repère de référence, il sera utilisé comme un paramètre de la pose de la plate-forme mobile
- x_C, y_C, z_C : coordonnées du point C dans le repère de référence
- A_i : centre de l'articulation de la i -ème jambe qui se situe sur la base du robot
- $x_{A_i}, y_{A_i}, z_{A_i}$: coordonnées du point A_i dans le repère de référence Ω_O
- B_i : centre de l'articulation de la i -ème jambe qui se situe sur la plate-forme mobile du robot
- $x_{B_i}, y_{B_i}, z_{B_i}$: coordonnées du point B_i dans le repère mobile Ω_C
- J^{-1} : jacobienne cinématique inverse
- J^{-T} : transposée de la jacobienne cinématique inverse

Analyse par intervalles

- \mathbb{IR} : l'ensemble des intervalles réels
- $X = [\underline{x}, \bar{x}]$: un intervalle et ses bornes inférieure et supérieure
- $w(X)$: la largeur de l'intervalle
- $mid(X)$: le centre ou point milieu d'un intervalle
- $rad(X)$: rayon de l'intervalle
- \mathbf{x}^I : un vecteur d'intervalles ou plus simplement une boîte
- \mathbf{A}^I : une matrice intervalle

- $f(\mathbf{x}^I)$: image de la fonction f sur la boîte \mathbf{x}^I
- $f^I(\mathbf{x}^I)$: extension intervalle de la fonction f sur la boîte \mathbf{x}^I

INTRODUCTION

1

1.1 Introduction

Le terme "Robot" apparaît au monde pour la première fois, en 1920 lors de la publication de la pièce de théâtre "*R.U.R : Rossum's Universal Robots*" de l'écrivain tchèque Karel Capek. Cependant ce dernier a toujours reconnu la paternité du mot à son frère aîné Josef. Même si le terme "*robota*", dont il est issu, existait déjà en slave et nous renvoie à la notion de corvée. Ainsi le robot aurait pour but initial de soulager l'homme de tâches lourdes, répétitives et faciliter son existence. Nous comprendrons que cette idée n'est pas nouvelle, et qu'elle a probablement jalonné le parcours scientifique de nombreux inventeurs depuis l'avènement de la science. Ainsi Aristote dans "*La Politique*" imagine un monde où chaque "instrument" serait capable, sur une simple injonction d'accomplir la tâche qui lui est propre, ce travail effectué par les "instruments" rendant l'esclavage obsolète. Une autre thématique de la robotique actuelle, à savoir l'aide à la réhabilitation se retrouve également dans les mythes antiques grecs. Selon la légende, Hephaïstos symbole de l'inventeur qui a souvent été représenté sous les traits d'un forgeron boiteux aurait bâti des servantes d'or pour l'aider à marcher.

Ainsi, afin de se soustraire à des tâches pénibles, répétitives ou dangereuses, l'Homme a donc de tout temps fantasmé ou créé des systèmes mécaniques. Ils consistent, pour nombres d'entre eux, à déplacer un solide (qu'on appelle plate-forme mobile, organe terminal en robotique) relativement à un corps fixe (appelé de manière générique base). Rappelons que le mouvement d'un solide dans l'espace peut se décomposer en 6 transformations géométriques : 3 translations (usuellement suivant trois axes orthogonaux) et trois rotations (autour de ces trois axes). Aussi pour décrire la mobilité d'un mécanisme on utilise le plus souvent la notion de degrés de liberté (mouvements indépendants possible de l'organe terminal relativement à la base). Si nous pouvons en commander plusieurs, nous parlerons alors de robot manipulateur.

Nous comptons deux grandes familles de robots industriels : les robots séries et les robots parallèles. La plus connue, mais aussi la plus ancienne est celle dite de série. Pour information le premier robot industriel, l'Unimate (figure 1.1) est vendu à General Motors en 1961 et c'est sur les chaînes de montage de la firme automobile américaine qu'il est pour la première fois utilisé. Ce type de robot a une structure plus ou moins proche de celle d'un bras humain. Il est constitué de segments (ou solides) reliés entre eux par des articulations à un degré de liberté. En robotique nous utilisons donc le plus souvent soit l'articulation prismatique notée P (une translation possible selon une direction), soit l'articulation rotoïde notée R (une rotation possible selon un axe). Nous trouverons également les liaisons rotule (notée S) et cardan (notée U).



FIG. 1.1: Le premier robot industriel : l'Unimate

Comme chaque segment, excepté la base et l'organe terminal, est relié à un seul segment successeur ou un seul segment prédécesseur, le terme de chaîne cinématique ouverte est utilisé (figure 1.2).

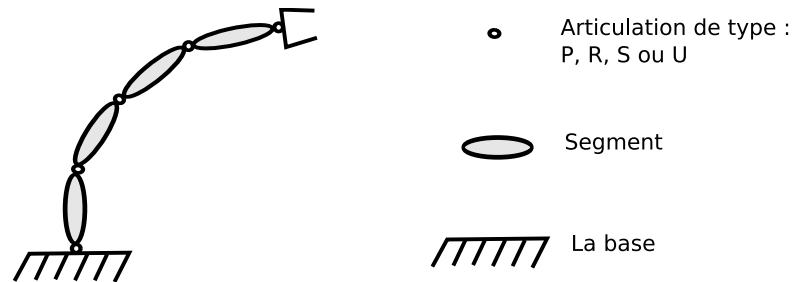


FIG. 1.2: chaîne cinématique ouverte

Cependant les robots séries sont, par nature, inadaptés pour réaliser certaines tâches. Par exemple, le transport de lourdes charges solliciterait de manière trop importante les segments proche de la base. Une solution permettant un meilleur équilibrage statique consiste à relier la base et l'organe terminal afin de répartir la charge sur les différentes chaînes cinématiques indépendantes. Ces chaînes cinématiques prises deux à deux forment des chaînes cinématiques fermées (figure 1.3). C'est ce qui définit les robots parallèles (figure 1.3).

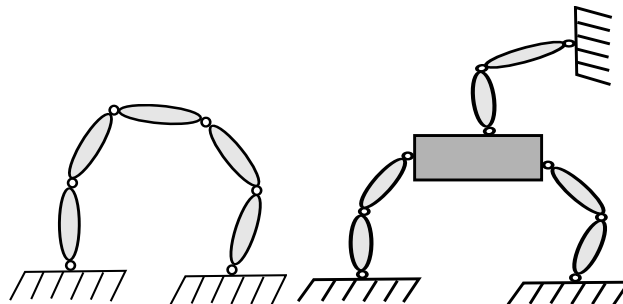


FIG. 1.3: chaîne cinématique fermée et robot parallèle

1.2 Les robots parallèles

1.2.1 Définition :

Définition : un robot parallèle est un mécanisme en chaîne cinématique fermée, constitué d'un organe terminal à n degrés de libertés et d'une base fixe, reliés entre eux par des chaînes cinématiques indépendantes. [35]

Nous traiterons dans ce document des robots parallèles ne possédant qu'un seul mode d'assemblage et la motorisation s'effectue par n actionneurs simples, un pour chaque jambe du mécanisme.

Depuis quelque temps déjà, la robotique parallèle a pris pied dans l'industrie, et ce dans de nombreux domaines comme l'usinage, la médecine (figure 1.4), le positionnement d'antennes ou de caméras [15], mais aussi dans le "*pick and place*" avec notamment le succès du Robot Delta [10] (figure 1.5). Il existe également des simulateurs de vol (figure 1.6), principalement construits sur la base d'une des plus célèbres architectures parallèles, la plate-forme de Gough [17], [50] (figure 1.7).



FIG. 1.4: Robot d'aide à la chirurgie de Siemens



FIG. 1.5: Robot Delta ABB FlexPicker

1.2.2 Avantages et inconvénients des robots parallèles

Les robots manipulateurs les plus répandus dans l'industrie restent sans aucun doute les robots séries. Cependant pour certains types d'applications, les robots parallèles se montrent, au vu des qualités intrinsèques de leur architecture, plus performants. Il est donc bon de comparer ces derniers à leurs cousins sériels afin d'identifier clairement



FIG. 1.6: Simulateur de vol Air France B777

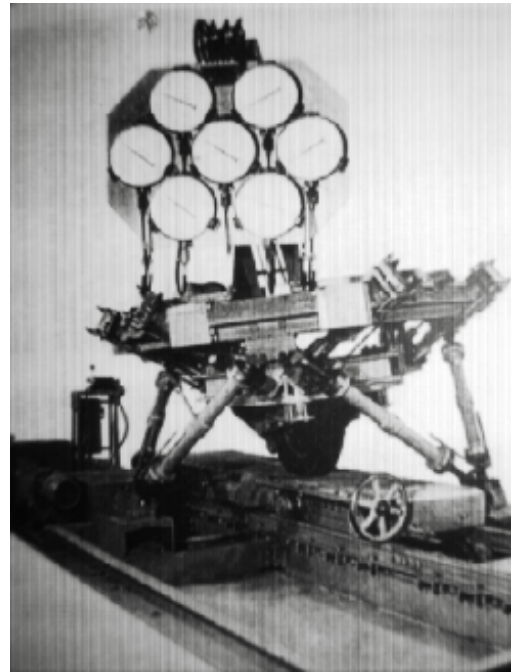


FIG. 1.7: La plate-forme de Gough (1947) utilisée pour tester des pneumatiques

quelle famille de robot est la plus adaptée pour une tâche donnée. Évidemment les performances d'un robot sont intimement liées à la géométrie, aux actionneurs et aux capteurs, aussi nous ne pourrions dégager que des principes généraux sur leurs avantages et leurs inconvénients.

Une des modélisations mathématiques utilisées en automatique pour le contrôle du robot considère chaque segment comme rigide, c'est-à-dire qu'il ne subit aucune déformation. Si nous considérons un robot série, nous nous apercevons que chaque segment doit supporter outre la charge placée sur l'organe terminal, celles des segments suivants. Ceci entraîne donc une déformation de type flexion. Et ces mêmes flexions (qui ne sont pas mesurées par les capteurs) induisent des erreurs de positionnement d'où l'intérêt de commande référencée vision.

Dans une architecture série il peut exister un autre facteur d'erreur de positionnement. En effet, une erreur de mesure d'un capteur sur le premier ou second segment va se répercuter de manière plus conséquente sur la pose de l'organe terminal. Aussi compte-tenu de l'architecture série, nous assistons à une accumulation/propagation des erreurs qui peut se révéler désastreuse dans certaines applications où nous rechercherons par exemple une forte précision.

Une solution possible pour limiter ces erreurs est d'augmenter la rigidité des segments mais cela passe par une augmentation conséquente de la masse du robot. Cela mène inévitablement à une baisse du rapport masse transportée sur masse du robot et à une augmentation conséquente des effets dynamiques nuisibles à la précision. Concernant un robot parallèle, le ratio va être meilleur car la charge est répartie entre les

différentes chaînes. Les flexions pourront parfois être négligées puisque généralement le mécanisme subit des déformations du type traction/compression qui se révèlent moins problématiques.

De plus, les erreurs de mesures dues aux capteurs sont réparties sur les chaînes cinématiques et donc l'erreur sur la pose de la plate-forme souffre moins de la répartition des erreurs contrairement à un mécanisme sériel.

Avec le même type de moteurs, un robot parallèle va pouvoir en combinant ses différentes jambes mais aussi compte-tenu de la masse moins élevée de l'équipage mobile, déplacer l'organe terminal plus rapidement. En effet, une réduction de la masse déplacée permet d'obtenir des caractéristiques dynamiques plus élevées.

Il est cependant illusoire de penser qu'un robot parfait puisse exister, aussi les structures parallèles n'ont bien évidemment pas que des avantages.

Ainsi, l'espace de travail d'un robot parallèle dépend des différentes chaînes cinématiques indépendantes qui relient la base à la plate-forme mobile. Aussi il est constitué de l'intersection des espaces de travail de chaque chaîne cinématique reliant la plate-forme à la base. En conséquence, l'espace de travail d'un robot parallèle est généralement réduit en raison des différentes contraintes imposées par les chaînes cinématiques qui relient la base du robot à la plate-forme mobile de ce dernier.

Un autre inconvénient dû aux contraintes des différentes jambes est l'existence de singularités (points où le mécanisme parallèle perd sa rigidité naturelle et voit parfois l'organe terminal perdre ou gagner des degrés de libertés rendant la plate-forme mobile incontrôlable). On comprend que ces poses particulières du robot sont donc d'un intérêt primordial aussi bien au niveau mécanique que pour le contrôle du robot, surtout si nous avons affaire à des mécanismes nécessitant une très forte précision (robotique chirurgicale par exemple).

1.3 Les singularités

Dans le langage courant, l'adjectif singulier fait référence à quelqu'un ou quelque chose d'unique, de rare, qui ne ressemble pas aux autres. Ainsi c'est par opposition à l'attitude normale, au comportement général des choses que nous allons désigner une entité singulière. Mathématiquement parlant, si nous imaginons qu'un ensemble quelconque est dirigé par une règle presque partout, c'est justement les éléments n'obéissant pas à cette règle qui vont définir le lieu singulier de l'ensemble.

Par exemple pour la valeur particulière 0 la fonction $f(x) = \frac{1}{x}$ n'est pas définie correctement et elle tend vers l'infini à son voisinage. Ainsi pour un robot parallèle, nous qualifions de lieu singulier, l'ensemble des positions pour lesquelles les règles établissant un comportement physique général du mécanisme ne peuvent être appliquées.

Il existe différents moyens d'introduire les singularités pour les robots parallèles. Nous avons choisi de les introduire en nous intéressant au comportement statique d'un mécanisme.

Soit \mathcal{F} l'effort extérieur (on prend donc en compte l'ensemble des forces et des moments) appliqué sur la plate-forme du manipulateur parallèle.

Soit τ l'ensemble des *forces articulaires* (forces/couples exercés sur les moteurs)

Nous verrons plus tard qu'il existe une matrice A dépendante de la *pose* χ du mécanisme telle que la relation fondamentale de la statique s'écrive :

$$A(\chi)\tau = \mathcal{F} \quad (1.1)$$

Aussi la résolution de l'équation (1.1) par la méthode de Cramer, nous permet d'exprimer chaque force articulaire τ_i comme :

$$\tau_i = \frac{|N_i|}{|A|} \quad \text{si } |A| \neq 0 \quad i = 1..3 \quad (1.2)$$

La matrice N_i s'obtient en substituant la i -ème colonne de A par \mathcal{F} .

Si le déterminant de la matrice A est nul, par définition nous dirons que la matrice A est singulière et comme A est dépendante de la pose χ , on parle alors de *pose singulière* χ_s . On abordera la terminologie propre aux singularités dans le chapitre 2. Pour une pose singulière, le système (1.1) devient un système linéaire sous-contraint qui peut admettre des solutions bornées. Ainsi d'un point de vue statique, la seule détermination des singularités n'est pas suffisante et il faut également s'intéresser au voisinage de ces configurations particulières. Comme l'indique l'équation (1.2), si les déterminant $|N_i|$ ne tendent pas vers 0 à l'approche d'une pose singulière χ_s alors nous pouvons dire que les forces articulaires τ_i tendent vers l'infini à l'approche de χ_s : ce qui conduit à une destruction du mécanisme !

Plusieurs problèmes découlent donc de cette constatation :

- Sommes-nous "loin" d'une singularité ?
- Vérifier pour une trajectoire donnée qu'il n'y a pas d'influence négative d'une singularité .
- Trouver une trajectoire qui évite la détérioration du mécanisme.
- Concevoir des mécanismes sans singularité.
- Déterminer des espaces sans influence des singularités.

A la lecture de ces différents problèmes, nous pouvons penser que les singularités des robots parallèles ne sont qu'un inconvénient majeur. Cependant on considérera que les singularités d'un point de vue statique ne sont dangereuses que lorsque les forces articulaires τ_i tendent vers l'infini. Ainsi si les matrices N_i ont leur déterminant qui tendent vers 0 à l'approche d'une position singulière, il peut exister des τ_i finies (voir l'équation (1.2)). Nous nuancerons donc le précepte qui veut que les singularités doivent être évitées. Cela est généralement vrai mais il existe des exceptions comme les mécanismes en *singularité permanente* [23], [26]. Pour ces mécanismes singuliers, les efforts subis par les articulations sont bornés, et leur trajectoire naturelle (qui peut-être parfois décrite avec un seul moteur) peut présenter un intérêt par exemple pour des applications liées à l'usinage de formes complexes.

1.4 Contributions

La liste de nos contributions est la suivante :

- *Contribution 1* : Nous définissons un *indice de proximité aux singularités qui repose sur un critère physique* : il s'agit de la valeur absolue des force/couples τ dans les jambes du robot. Nous comprendrons qu'il doit rester inférieur à un seuil donné τ_{max} qui correspond dans le cas limite au seuil de rupture du plus faible composant d'une jambe. Cependant nous pourrions considérer τ_{max} comme le seuil d'effort admissible qui est toujours inférieur au seuil de rupture car il prévient les cas où les composants des jambes peuvent subir une déformation non élastique. Nous retrouverons cette indice dans le chapitre 3.

- *Contribution 2* : Nous définissons *l'espace de travail statique* W_s comme l'ensemble des poses où l'indice est inférieur à τ_{max} . C'est un espace sûr au sens de la statique puisqu'il permet d'éviter d'avoir des forces/couples dans les jambes excédant le seuil de rupture. Ainsi dans W_s l'intégrité du mécanisme est assurée. Nous retrouvera la définition de l'espace de travail statique dans le chapitre 3.

- *Contribution 3* : Nous proposons deux algorithmes permettant de calculer l'espace de travail statique.

Algorithme 1 : il repose sur une approche algébrique qui permet de calculer la frontière de coupe plane de l'espace de travail statique (chapitre 3).

Algorithme 2 : utilisant une approche analyse par intervalles, il permet de calculer une approximation de l'intégralité de l'espace de travail statique, quelle que soit la dimension de l'espace dans lequel il est immergé (chapitre 5). De plus nous verrons que sous réserves de modifications mineures de notre algorithme, nous sommes capables de répondre à différents problèmes ouverts sur les singularités : vérifier qu'une zone de l'espace de travail est "loin" d'une singularité, vérifier qu'une trajectoire ne risque pas de détériorer le mécanisme, prendre en compte les incertitudes liées à la fabrication du robot et vérifier qu'une zone est sûre au sens statique, aider à la conception de mécanismes où les singularités ne menacent pas l'intégrité du robot.

- *Contribution 4* : Nous avons proposé des observations sur l'analyse par intervalles, qui permettent d'améliorer un certain type de méthodes de résolutions que l'on nomme "*shaving*" ou rognage dans la littérature. Une introduction à ce type de méthodes dite aussi de *filtrage* se trouve dans le chapitre 4, et nous détaillerons nos observations et leurs impacts dans les différentes briques nécessaires à la réalisation d'un algorithme de résolution de systèmes intervalles dans le chapitre 5.

NOTATIONS, PRÉREQUIS ET SINGULARITÉS

Ce chapitre va nous permettre d'introduire les différentes notations liées à la robotique parallèle, ainsi que les éléments nécessaires à une bonne compréhension du phénomène que représente les singularités.

2.1 Notations

2.1.1 Nomenclature d'un mécanisme

En premier lieu, afin de décrire un mécanisme en robotique parallèle, nous nous restreignons aux types d'articulations suivantes :

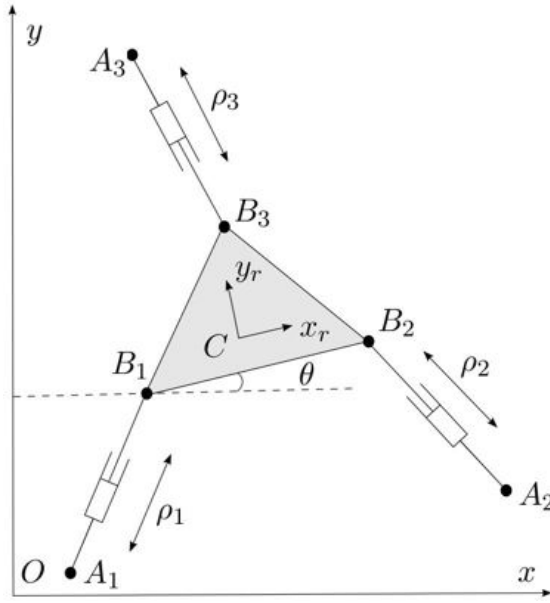
- l'articulation *prismatique* (notée P) permet une translation suivant une direction.
- l'articulation *rotoïde* (notée R) permet une rotation suivant un axe.
- l'articulation *cardan* (notée U) permet une rotation suivant deux axes.
- l'articulation *rotule* (notée S) permet une rotation suivant trois axes.

Pour décrire la structure mécanique d'une jambe de robot parallèle, nous utilisons communément la liste de ces articulations à partir de la base. Par exemple la notation RPR indique que partant de la base la première articulation est une rotoïde qui sera suivie d'une prismatique elle même suivie d'une rotoïde qui sera connectée à la plate-forme. De plus pour différencier une articulation *active* (articulation qui peut-être contrôlée directement par l'utilisateur, par exemple via un moteur) d'une articulation passive, la notation de l'articulation active sera soulignée. Ainsi une articulation prismatique dont nous pouvons contrôler la longueur grâce à un moteur sera notée \underline{P} . Si le mécanisme possède des jambes identiques, nous pouvons alléger la notation en faisant précéder une structure de jambe par le nombre de chaîne de ce type. Ainsi la notation $2 - R\underline{P}R - \underline{R}R\underline{R}$ désigne un robot à 3 chaînes, 2 de type $R\underline{P}R$ et une de type $\underline{R}R\underline{R}$.

2.1.2 Notations

Nous allons désormais introduire les notations qui vont être utilisées tout au long de la thèse sur le mécanisme plan 3 – $R\underline{P}R$: c'est un mécanisme qui possède donc trois jambes avec à la base une articulation rotoïde, reliée par une articulation prismatique active à une dernière articulation rotoïde liée à la plate-forme (figure 2.1).

Nous définissons :

FIG. 2.1: Le robot 3 – \underline{RPR}

- un repère de référence Ω_O ($O, \mathbf{x}, \mathbf{y}, \mathbf{z}$) lié à la base du robot.
- un repère mobile Ω_C ($C, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r$) lié à la plate-forme du manipulateur.
- le vecteur \mathbf{OC} défini dans Ω_O permet de représenter la position du mobile. On désignera la position par le vecteur \mathbf{X} .
- l'angle θ entre l'axe x du repère de référence et l'axe x_m du repère mobile représente l'orientation de la plate-forme pour notre robot plan. Cependant si nous traitons un robot possédant 3 degrés de libertés en rotation comme la plate-forme de Gough par exemple nous désignerons par ζ l'ensemble des paramètres désignant la rotation. Une des représentations les plus utilisées est celle des angles d'Euler.
- Nous désignons par *coordonnées généralisées* $\chi = (\mathbf{X}, \zeta)$ l'ensemble des paramètres qui permettent de caractériser la pose de la plate-forme mobile.
- i l'indice définissant une des jambes du robot.
- A_i le centre de l'articulation liant la base à la jambe i . Ses coordonnées \mathbf{OA}_i sont données dans le repère de référence.
- B_i le centre de l'articulation liant la plate-forme mobile à la jambe i . Ses coordonnées \mathbf{CB}_{ir} sont exprimées dans le repère mobile.
- Les coordonnées des points A_i et B_i (respectivement dans le repère de référence et dans le repère mobile) font partie des *paramètres géométriques du robot* ψ , qui comme leur nom l'indique permettent de décrire la géométrie du mécanisme.
- ρ_i désigne la longueur des jambes, ce qui correspond pour le 3 – \underline{RPR} à la longueur de l'articulation prismatique. Les ρ_i font parties des *coordonnées articulaires* Θ c'est à dire les variables qui permettent d'établir la position des articulations. On distinguera les coordonnées articulaires actives Θ_a (ici les ρ_i) des passives Θ_p (ici les angles des articulations rotoïdes).

2.2 Modèles géométriques

Nous désignons par modèle géométrique un système d'équations qui décrit les contraintes géométriques liées à la structure particulière d'un mécanisme étudié. Ce système d'équations lie les paramètres géométriques du robot ψ , les coordonnées généralisées χ (autrement dit la pose de la plate-forme mobile) et les coordonnées articulaires Θ . Ces équations sont des briques élémentaires qui sont nécessaires pour le calcul d'une trajectoire, pour l'étalonnage d'un mécanisme ou encore pour sa modélisation. Nous distinguerons deux types de modèles géométriques : le modèle géométrique inverse (MGI) et le modèle géométrique direct (MGD). Précisons cependant qu'il s'agit d'un abus de langage, certains préfèrent utiliser un seul modèle géométrique (le modèle géométrique implicite) et désignent les différentes manières de résoudre le modèle par problème géométrique inverse et problème géométrique direct.

2.2.1 Modèle géométrique inverse

Le MGI consiste à obtenir la valeur des coordonnées articulaires Θ correspondant à la valeur des coordonnées généralisées χ .

Il est usuellement facile à obtenir pour les robots parallèles tant que la géométrie des jambes reliant la base à la plate-forme mobile reste assez simple pour une analyse mathématique. Nous commençons par exprimer les vecteurs $\mathbf{A}_i\mathbf{B}_i$ en fonction des coordonnées généralisées χ .

$$\mathbf{A}_i\mathbf{B}_i = H_1(\chi) \quad (2.1)$$

Comme nous souhaitons déterminer la valeur des différentes articulations (parfois seules les articulations actives suffisent), on exprime ensuite les vecteurs $\mathbf{A}_i\mathbf{B}_i$ comme des fonctions des coordonnées articulaires et, si nécessaire, des coordonnées généralisées.

$$\mathbf{A}_i\mathbf{B}_i = H_2(\chi, \Theta) \quad (2.2)$$

En combinant les équations (équation 2.1) et (équation 2.2), on obtient $H_1(\chi) = H_2(\chi, \Theta)$ donc une relation du type :

$$F(\Theta, \chi) = 0 \quad (2.3)$$

Si la géométrie est simple, il est parfois possible d'obtenir alors $\Theta = G(\chi)$

Exemple : MGI du robot plan 3 – RPR

En utilisant la relation de Chasles, on obtient :

$$\begin{aligned} \mathbf{A}_i\mathbf{B}_i &= \mathbf{A}_i\mathbf{O} + \mathbf{OC} + \mathbf{CB}_i \\ &= \mathbf{OC} + R\mathbf{CB}_{ir} - \mathbf{OA}_i \end{aligned}$$

où R est la matrice de rotation qui permet d'obtenir les coordonnées d'un vecteur dans le référentiel absolu Ω_O à partir du vecteur exprimé dans le repère mobile Ω_C . Comme la longueur ρ_i d'une jambe i est la norme du vecteur $\mathbf{A}_i\mathbf{B}_i$, nous pouvons écrire :

$$\rho_i = \|\mathbf{A}_i\mathbf{B}_i\| = \|\mathbf{OC} + R\mathbf{CB}_{ir} - \mathbf{OA}_i\|$$

Au final, on obtient 3 équations à 3 inconnues ($\chi = (x_c, y_c, \theta)$) mais il est souvent plus pratique d'utiliser le carré de la longueur d'une jambe soit : $\rho_i^2 = \|\mathbf{OC} + R\mathbf{CB}_{ir} - \mathbf{OA}_i\|^2$

2.2.2 Modèle géométrique direct

Le MGD est le problème inverse du MGI, il consiste donc à obtenir les valeurs des coordonnées généralisées χ correspondant aux valeurs fixées des coordonnées articulaires actives Θ_a .

Pour contrôler les mouvements d'un robot parallèle, il est nécessaire de déterminer la position et l'orientation de l'organe terminal en fonction des longueurs des jambes qui sont plus facilement mesurables que la pose de la plate-forme. Contrairement au MGI, le MGD est généralement complexe et souvent il admet plusieurs solutions. Cela signifie qu'il n'existe pas de solution unique pour assembler un mécanisme parallèle en ne prenant pas en compte les articulations passives (qui par définition ne peuvent pas être directement contrôlées par l'utilisateur). Les différentes solutions du MGD sont appelées les *modes d'assemblage*.

A titre d'informations, pour les robots plans du type 3 – *RPR* il existe jusqu'à six modes d'assemblages [14] alors que la plate-forme de Gough peut en compter jusqu'à quarante [12].

2.3 Modèles cinématiques

Avant de définir de manière succincte les modèles cinématique direct (MCD) et indirect (MCI), nous allons d'abord introduire quelques notations qui vont permettre la description de ces modèles.

Nous noterons les *vitesses articulaires* par $\dot{\Theta}$, les *vitesses des articulations actives* par $\dot{\Theta}_a$ et les *vitesses des articulations passives* par $\dot{\Theta}_p$.

Nous rappelons également la définition du *torseur cinématique* \mathcal{V} appliqué au point C et exprimé généralement dans le repère absolu. Le torseur cinématique est composé de la *vitesse cartésienne au point C*, \mathbf{V} , et du *vecteur des vitesses angulaires* Ω . Nous définirons le torseur cinématique relatif aux n degrés de liberté du robot par \mathcal{V}_n et le complémentaire de \mathcal{V}_n dans \mathcal{V} sera noté \mathcal{V}_e .

Définitions :

- Le MCI consiste à déterminer les vitesses articulaires en fonction du torseur cinématique \mathcal{V} d'un point C de la plate-forme mobile.
- Le MCD consiste pour sa part à déterminer \mathcal{V} en fonction de $\dot{\Theta}_a$.

Il existe plusieurs méthodes pour obtenir le MCI. Ces dernières servent à l'étude analytique des singularités, et nous allons maintenant les rappeler.

Un moyen d'obtenir les modèles cinématiques consiste à dériver les équations du MGI dont nous avons vu (équation 2.3) qu'il s'obtient sous la forme $F(\chi, \Theta) = 0$.

Si Θ est réduit aux n articulations actives Θ_a , on obtient par différentiation une relation linéaire d'entrée-sortie pour la vitesse :

$$A(\chi, \Theta)\dot{\Theta}_{\mathbf{a}} + B(\chi, \Theta)\mathcal{V} = 0 \quad (2.4)$$

Si nous avons au plus un degré de liberté en rotation il y a identité entre $\dot{\chi}$ et \mathcal{V} . Par conséquent nous pouvons donc écrire le MCI sous la forme :

$$\dot{\Theta}_{\mathbf{a}} = J^{-1}\dot{\chi} \quad \text{avec} \quad J^{-1} = -A^{-1}B \quad (2.5)$$

Le terme J^{-1} est appelé *jacobienne cinématique inverse* par cohérence avec la notation utilisé pour les robots série, pour lesquels nous obtenons facilement $\dot{\chi} = J\dot{\Theta}$, J étant appelée jacobienne cinématique.

Cependant si il y a plus d'un degré de liberté en rotation l'égalité entre le torseur cinématique \mathcal{V} et la dérivée des coordonnées généralisées $\dot{\chi}$ est perdue. Montrons le :

Choisissons par exemple la représentation de l'orientation dite des angles d'Euler :

- On passe du repère d'origine $0xyz$ au repère $Ouvz$ par une rotation d'angle ψ (angle de précession) autour de l'axe z .
- On passe du repère $Ouvz$ au repère $Ouwz'$ par une rotation d'angle θ (angle de nutation) autour de l'axe u .
- On passe du repère $Ouwz'$ au repère $x'y'z'$ par une rotation d'angle ϕ (angle de rotation propre) autour de l'axe z' .

Le vecteur des vitesses angulaires s'exprime alors par $\Omega = \dot{\psi}z + \dot{\theta}u + \dot{\phi}z'$. Si la relation est retranscrite dans la base initiale, nous obtenons :

$$\Omega = \begin{pmatrix} 0 & \cos(\psi) & \sin(\psi)\sin(\theta) \\ 0 & \sin(\psi) & -\cos(\psi)\sin(\theta) \\ 1 & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix}$$

De manière plus générale et pour une représentation quelconque de l'orientation, nous pouvons définir une matrice inversible H telle que :

$$\mathcal{V} = H\dot{\chi} \quad \text{ou} \quad \dot{\chi} = H^{-1}\mathcal{V} \quad (2.6)$$

Pour les angles d'Euler H sera usuellement inversible sauf pour des orientations particulières (par exemple si l'angle de nutation θ est nul, les angles ψ et ϕ décrivent des rotations autour du même axe et donc seule leur somme est significative) et nous parlons alors de *singularité de représentation*.

Nous définissons le *modèle cinématique minimal*, comme celui n'impliquant que $\mathcal{V}_{\mathbf{n}}$ et $\dot{\Theta}_{\mathbf{a}}$. Le modèle cinématique minimal s'écrira donc : $\dot{\Theta}_{\mathbf{a}} = J^{-1}\mathcal{V}_{\mathbf{n}}$ mais il lui est parfois préféré la formulation suivante lors de l'étude des singularités :

$$A(\chi, \Theta)\dot{\Theta}_{\mathbf{a}} + B_k(\chi, \Theta)\mathcal{V}_{\mathbf{n}} = 0 \quad \text{avec} \quad B_k = BH^{-1} \quad (2.7)$$

Il convient de mentionner que pour une étude complète des singularités, il faut toujours considérer les six degrés de libertés de la plate-forme, même pour des robots ayant a priori moins de six degrés de liberté, sous peine de ne pas identifier toutes les singularités (voir l'exemple du robot 3-*UPU* [51]). Il sera donc parfois nécessaire de considérer les

articulations passives comme des variables à différencier dans l'équation du MGI (équation 2.3), ce qui nous conduit au système d'équations suivant appelé *modèle cinématique complet* :

$$A\dot{\Theta}_a + BH^{-1}\mathcal{V} + C\dot{\Theta}_p = 0 \quad (2.8)$$

2.4 Singularités des robots parallèles

Nous avons déjà introduit les singularités dans le cadre d'une approche statique mais nous allons maintenant voir que ce sont aussi des configurations particulières où le mécanisme, qui est usuellement rigide avec ses actionneurs bloqués, va exhiber localement des mobilités. Nous comprendrons aisément qu'une perte de contrôle due aux singularités, est inacceptable quelque soit l'application. En perdant sa rigidité, les forces dans les jambes ou les couples développés par les moteurs peuvent se révéler très importants ce qui peut entraîner de sérieuses avaries voire la rupture du mécanisme.

C'est pourquoi il a d'abord été cherché à éviter les singularités en recherchant des trajectoires de contournement [3], [42], [45], [49]. Mais pour les éviter il faut avant tout les identifier. Bien que les premiers travaux sur les singularités des robots parallèles soient l'oeuvre de Borel [7], Bricard [8] et Cauchy [9], le pionnier de l'étude des singularités pour l'ère moderne fut Hunt [28] qui étudia le phénomène en s'attachant aux paramètres géométriques du mécanisme. Ainsi il introduisit les concepts de “*uncertainty configuration*” et de “*stationnary configuration*” en étudiant des mécanismes 4-barres comme le *RRRR*.

La première étude de référence sur les singularités des robots parallèles fut établie par Gosselin et Angeles [13] qui utilisent le modèle cinématique minimale (équation 2.7).

Avec ce modèle, ils vont distinguer 3 types de singularités :

type 1 : La matrice A_k est singulière (donc son déterminant est nul). Dans cette configuration, il existe un vecteur $\dot{\Theta}_a$ non nul (au moins un des actionneurs est actif) pour lequel la plate-forme mobile ne bouge pas ($\mathcal{V}_n = 0$) : on parle alors de *singularité sérielle*. Il est souvent indiqué qu'une telle singularité correspond à une frontière de l'espace de travail mais l'étude du robot 3 – *PRP* a montré que ceci n'était pas exact [6]. Dans une singularité de type 1, le robot perd la capacité de déplacer l'organe terminal dans une direction précise, on dit alors qu'il perd un degré de liberté.

type 2 : La matrice B_k est singulière. Dans une telle position, le torseur cinématique \mathcal{V}_n peut-être non nul bien que les vitesses articulaires soient nulles ($\dot{\Theta}_a = 0$). Cela signifie simplement que la plate-forme mobile peut être en mouvement alors qu'aucun des moteurs n'est actif. Dans le voisinage d'une singularité de type 2, le robot gagne un ou plusieurs degrés de liberté : puisqu'il est susceptible de bouger alors que ces actionneurs sont bloqués. Un ou plusieurs des degrés de liberté deviennent incontrôlable. Cela peut porter à de graves conséquences, surtout dans le cadre d'applications requérant une forte précision. Les utilisateurs et concepteurs cherchent donc le plus souvent à éviter

ce type de singularités.

type 3 : Les matrices A_k et B_k sont singulières, la plate-forme du robot peut se mouvoir alors que tous les moteurs sont à l'arrêt. Cependant il existe certaines directions où il est impossible de déplacer la plate-forme mobile malgré des moteurs actifs.

Cette classification s'est avérée insuffisante, comme le prouve l'étude menée par Zlatanov sur le robot 3- UPU [54]. En ne prenant en compte que le torseur cinématique restreint aux degrés de liberté de la plate-forme mobile, nous pouvons être amené à ignorer des singularités pour les manipulateurs parallèles à mobilité restreinte (manipulateur comptant moins de six degrés de liberté). Aussi Zlatanov a étudié de manière plus générale les singularités en prenant en compte le torseur cinématique complet et les articulations passives en partant du modèle cinématique complet (équation 2.8), qui peut également s'écrire sous la forme :

$$L(\dot{\Theta}_a, \dot{\Theta}_p, \mathcal{V})^T = 0 \quad (2.9)$$

avec L , une matrice $N \times (N + n)$ pour un robot non redondant à n degrés de liberté et qui possède un total de N articulations. Zlatanov a défini, lui aussi, différents types de singularités [55].

Redundant input singularity (RI) : Elle est semblable à la singularité de type 1 de Gosselin : le torseur cinématique peut-être nul alors qu'il existe un actionneur en marche. Dans ce cas, le MCI ne peut pas être résolu.

Redundant output singularity (RO) : La plate-forme mobile exhibe un mouvement alors que tous les actionneurs sont bloqués. Nous pourrions penser qu'il s'agit de la singularité de type 2 définie par Gosselin et Angeles mais sa définition est plus large compte tenu de la prise en compte des articulations passives et du torseur cinématique complet. En effet pour la singularité de type 2 définie précédemment il suffit d'avoir $\mathcal{V}_n \neq 0$ alors qu'avec une RO, nous pouvons nous trouver dans le cas $\mathcal{V}_n = 0$ et $\mathcal{V}_e \neq 0$ ce qui correspond bien à $\mathcal{V} \neq 0$. Ce dernier cas qui ne peut être observé si on étudie seulement les équations d'entrée sortie du problème est appelé *singularité de contrainte* par Bonev et Zlatanov [5]. Précisons que dans le cas d'une RO, le modèle cinématique direct ne peut pas être résolu.

Redundant passive motion (RPM) : Cette singularité est observée quand une articulation passive est en mouvement ($\dot{\Theta}_p \neq 0$) alors que les actionneurs sont bloqués ($\dot{\Theta}_a = 0$) et la plate-forme mobile reste fixe ($\mathcal{V} = 0$). Park et Kim, qui utilisent la géométrie différentielle, les appellent aussi les *singularités d'actionneurs* [46].

2.5 Géométrie de Grassmann

La géométrie des droites, dite géométrie de Grassman [44], joue un rôle important dans l'étude classique des singularités. Pour l'introduire il est nécessaire de définir des concepts de base.

2.5.1 Vecteur de Plücker

Définition : Soit M_1 et M_2 , deux points appartenant à une même droite. Définissons le vecteur de Plücker P de dimension 6 et le vecteur de Plücker normalisé P_n par :

$$P = (\mathbf{M}_1\mathbf{M}_2, \mathbf{OM}_1 \times \mathbf{M}_1\mathbf{M}_2) \quad P_n = \left(\frac{\mathbf{M}_1\mathbf{M}_2}{\|\mathbf{M}_1\mathbf{M}_2\|}, \frac{\mathbf{OM}_1 \times \mathbf{M}_1\mathbf{M}_2}{\|\mathbf{M}_1\mathbf{M}_2\|}, \right)$$

Ces vecteurs définissent de manière unique une droite de l'espace Euclidien.

Remarques :

- la définition d'une droite par un vecteur de Plücker avec 6 composantes est redondante car seuls 4 paramètres sont nécessaires pour décrire une droite dans l'espace. Pour réduire cette redondance, le vecteur de Plücker normalisé a été défini (la normalisation fournissant une équation de contrainte).
- on peut réécrire P en utilisant les deux vecteurs de dimension 3 $\mathbf{M}_1\mathbf{M}_2 = p$ et $\mathbf{OM}_1 \times \mathbf{M}_1\mathbf{M}_2 = q$ comme $P = (p, q)$. Une condition nécessaire pour qu'un vecteur de dimension 6 soit un vecteur de Plücker (donc définisse une droite) est $p.q=0$.
- deux droites \mathcal{D}_1 et \mathcal{D}_2 sont sécantes si et seulement si $p_1.q_2 + q_1.p_2 = 0$.
- remarquons que les composantes d'un vecteur de Plücker, qu'il soit normalisé ou non, ne sont pas homogènes en terme d'unité.

2.5.2 Jacobienne cinématique inverse

Pour un grand nombre de robots parallèles, on peut exprimer la jacobienne inverse à l'aide de vecteurs de Plücker normalisés ou non [37]. A titre d'exemple nous allons calculer la matrice jacobienne de la plate-forme de Gough (figure 2.2), robot à six degrés de liberté.

Pour ce robot, le modèle géométrique implicite peut-être décrit par l'équation suivante

$$\rho^2 = \|\mathbf{AB}\|^2 = \|\mathbf{AO} + \mathbf{CB} + \mathbf{OC}\|^2 \quad (2.10)$$

En différenciant cette équation on obtient :

$$\rho\dot{\rho} = \mathbf{V}_c.\mathbf{AB} + \dot{\mathbf{CB}}.\mathbf{AC} \quad (2.11)$$

La relation entre le vecteur des vitesses de rotations Ω et $\dot{\mathbf{CB}}$ est donnée par :

$$\dot{\mathbf{CB}} = \mathbf{BC} \wedge \Omega \quad (2.12)$$

En réinjectant ce résultat dans l'équation (2.11), nous pouvons écrire :

$$\dot{\rho} = \frac{\mathbf{AB}}{\|\mathbf{AB}\|}.\mathbf{V}_c + (\mathbf{BC} \wedge \Omega).\frac{\mathbf{AC}}{\|\mathbf{AB}\|} = \frac{\mathbf{AB}}{\|\mathbf{AB}\|}.\mathbf{V}_c + \left(\frac{\mathbf{CB} \wedge \mathbf{AC}}{\|\mathbf{AB}\|}\right).\Omega \quad (2.13)$$

En posant ensuite $\mathbf{n}_i = \frac{\mathbf{AB}}{\|\mathbf{AB}\|}$ tout en considérant que $\mathbf{CB} \wedge \mathbf{AC} = \mathbf{CB} \wedge (\mathbf{AC} + \mathbf{CB})$, nous pouvons dire qu'une ligne de la jacobienne inverse J^{-1} va s'écrire sous la forme du

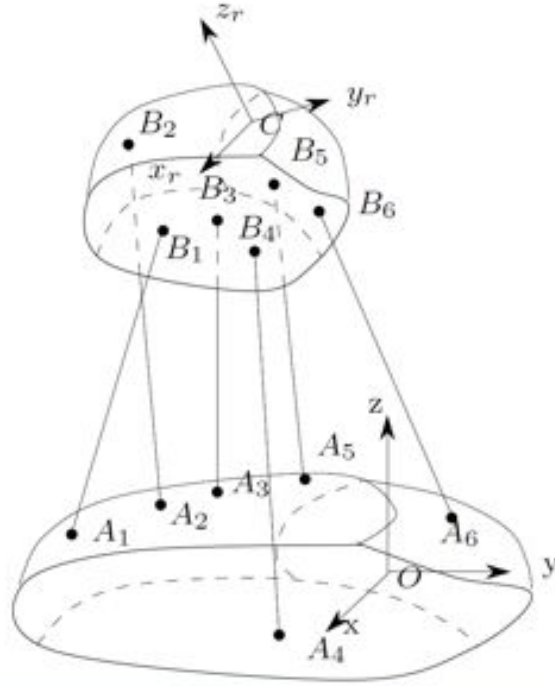


FIG. 2.2: La plateforme de Gough

vecteur de Plücker

$$(J^{-1})_i = (\mathbf{n}_i, \mathbf{CB}_i \wedge \mathbf{n}_i) \quad (2.14)$$

Remarque : En règle générale pour un robot spatial (respectivement plan) dont les jambes sont attachées à la plate-forme mobile par une articulation du type rotule (respectivement rotoïde), la jacobienne cinématique inverse est composée des vecteurs de Plücker des droites associées au dernier segment du robot (c'est-à-dire celui qui lie la jambe à l'organe terminal) [35].

2.5.3 Modèle statique

Soit le vecteur des forces articulaires τ , c'est à dire les forces exercées par les différentes jambes sur l'organe terminal. On désigne par le torseur des efforts extérieurs \mathcal{F} , l'ensemble des forces \mathbf{F} et des moments exercés \mathbf{M} sur un point C de la plate-forme mobile que nous considérons comme l'origine du repère mobile Ω_C . Notons également \mathbf{n}_i , le vecteur directeur normalisé ($\|\mathbf{n}_i\| = 1$) de la droite associée au dernier segment du robot. Le modèle statique consiste à établir la relation entre τ et \mathbf{F} .

Si nous considérons le système à l'équilibre :

$$\begin{cases} \sum_{i=1}^{i=n} \tau_i \mathbf{n}_i = \mathbf{F} \\ \sum_{i=1}^{i=n} \mathbf{CB}_i \wedge \tau_i \mathbf{n}_i = \mathbf{M} \end{cases} \quad (2.15)$$

qui peut être réécrit sous la forme du système linéaire suivant : $J^{-T}\tau = \mathcal{F}$ où J^{-T} est la transposée de la jacobienne cinématique inverse.

Remarque : Comme il existe une dualité entre la statique et la cinématique, les matrices impliquées dans les modèles sont les mêmes à une transposition près.

On sait que le déterminant d'une matrice est le même que celui de sa transposée. Par conséquent nous pourrions dire qu'une pose est singulière si et seulement si il existe une dépendance linéaire entre les équations du système (2.15), c'est-à-dire une dépendance linéaire entre les vecteurs de Plücker.

2.5.4 Singularités et géométrie de Grassmann

Nous avons vu dans la section précédente que pour la plupart des robots parallèles, les lignes de la matrice jacobienne cinématique inverse étaient des vecteurs de Plücker normalisés ou pas. Aussi si la matrice est singulière, cela signifie qu'il y a une dépendance linéaire entre ces vecteurs de Plücker. La géométrie de Grassman (XIX ème) montre que la dépendance linéaire entre les vecteurs de Plücker implique des relations géométriques entre les droites correspondantes [11], [47].

Par exemple pour le cas plan avec un $3 - RPR$, il y a dépendance linéaire des vecteurs de Plücker quand les droites qui leur sont associées ont un point commun. A gauche sur la figure 2.3 on peut observer le cas de trois droites concourantes en un point : il y a dépendance linéaire entre les droites donc le manipulateur est en pose singulière. A droite les trois droites sont parallèles : dans le cadre de la géométrie de Grassman il est considéré que ces droites ont un point commun à l'infini, nous avons donc affaire à une singularité.

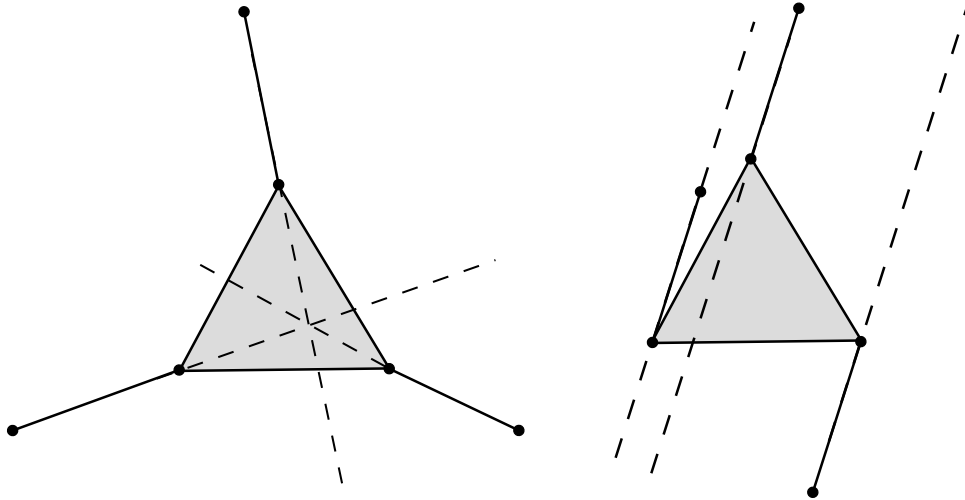


FIG. 2.3: Exemple de positions singulières d'un manipulateur parallèle plan $3 - RPR$

Le cas plan reste très simple mais pour les cas où $n > 3$, il vaut mieux se référer à des ouvrages plus complets. Nous conseillerons [37] pour une introduction à la géométrie de Grassman et ses applications à l'étude des singularités pour la robotique parallèle.

Insistons également sur l'intérêt que peut amener l'algèbre de Grassman-Cayley et la théorie des visseurs.

2.6 Transition

Comme nous l'avons déjà abordé dans le premier chapitre, connaître le lieu des singularités n'est pas suffisant car les variations de performances au voisinage d'une singularité peuvent se révéler très importantes et peuvent conduire à menacer l'intégrité du manipulateur parallèle, voire causer sa destruction. Il est donc intéressant de pouvoir répondre à la question : "Cette position est-elle sûre pour le robot ?" ce qui revient à se demander si l'on se situe assez "*loin*" des singularités. Dans le prochain chapitre, nous allons justement définir un indice de proximité aux singularités qui repose sur un critère physique ce qui nous permettra ensuite de définir un espace sûr (au sens statique) pour lequel l'intégrité physique du manipulateur est assurée.

ESPACE DE TRAVAIL STATIQUE DU 3-RPR

3.1 Introduction

Comme nous l'avons dit précédemment, lorsqu'un robot parallèle se trouve dans une position singulière, son comportement est modifié aussi bien cinématiquement que statiquement parlant. Cependant, même à l'approche d'une configuration singulière le mécanisme peut exhiber d'importantes variations de performances. Il semble donc opportun de définir un indice N qui établisse une notion de "*proximité*" à une position singulière.

Voglewede [52] propose trois conditions que doit respecter l'indice N :

- *condition 1* : $N(\chi) = 0$ si et seulement si pour la pose χ le mécanisme est singulier
- *condition 2* : Si χ n'est pas une pose singulière alors $N(\chi) > 0$
- *condition 3* : $N(\chi)$ doit avoir un sens physique.

Avant de s'attaquer aux possibles définitions d'un tel indice, il est important de rappeler que la jacobienne cinématique inverse J^{-1} n'est pas homogène en terme d'unités. Toute définition de N reposant sur des notions d'algèbre linéaire (norme, conditionnement, etc...) conduira à des valeurs qui seront dépendantes des unités utilisées (puisque rotation et translation sont traitées de la même manière). Cela n'est pas acceptable puisque cela revient à poser de manière arbitraire une relation d'équivalence entre rotation et translation [36].

Afin d'éviter de se heurter à ce problème d'homogénéité, lorsque nous étudions des robots possédant des degrés de libertés à la fois en translation et en rotation, nous garderons donc à l'esprit le principe de Merlet [38] qui rajoute une quatrième condition à celles données précédemment par Voglewede pour l'indice N :

- *condition 4* : N se doit d'être invariant quelles que soient les unités.

De nombreux travaux ont été menés pour déterminer le lieu des singularités [4], [33] ou pour les éviter [1], [24], [42], [45], [49]. Une solution pratique consiste à déterminer si un espace de travail donné ne contient pas de singularités tout en tenant compte des incertitudes dans la modélisation du robot [39]. Cependant l'idée de contourner ou de traverser une singularité repose sur une analyse cinématique qui ne prend pas en compte la valeur des forces/couples subis par les articulations et les moteurs. La première contribution de ce travail repose sur un critère physique très simple : pour

éviter toute détérioration du robot, il faut que les efforts dans les jambes de ce dernier soient inférieurs en valeur absolue à un seuil donné. Ce seuil correspond à la force ou au couple d'admissibilité du plus "faible" composant mécanique d'une jambe. Soit τ le vecteur des forces/couples exercés par les jambes et $\tau_{max} > 0$ la plus grande valeur admissible par le mécanisme n'entraînant pas de dégradations. Nous définissons *l'espace de travail statique* W_S comme une zone sûre (au sens statique) de l'espace de travail, c'est-à-dire l'ensemble des poses χ telles que :

$$\forall i \quad |\tau_i(\chi)| \leq \tau_{max} \quad (3.1)$$

Il s'ensuit la définition d'un indice N qui respecte trois des quatre conditions sur W_s .

$$N(\chi) = \min_{i=1,n} (\tau_{max} - |\tau_i(\chi)|) \quad (3.2)$$

En effet, si χ_0 est une pose telle que $N(\chi_0) = 0$ alors rien ne nous permet d'affirmer qu'elle est singulière. Cependant notre but n'est pas de caractériser si une pose est singulière mais plutôt de savoir si notre pose est dangereuse du point de vue statique c'est à dire à caractériser la frontière de l'espace de travail statique. Si jamais l'indice est négatif pour une pose donnée, nous pouvons conclure que cette pose est exclue de l'espace de travail statique et si au contraire l'indice est positif alors on peut affirmer que cette pose appartient à l'espace de travail statique.

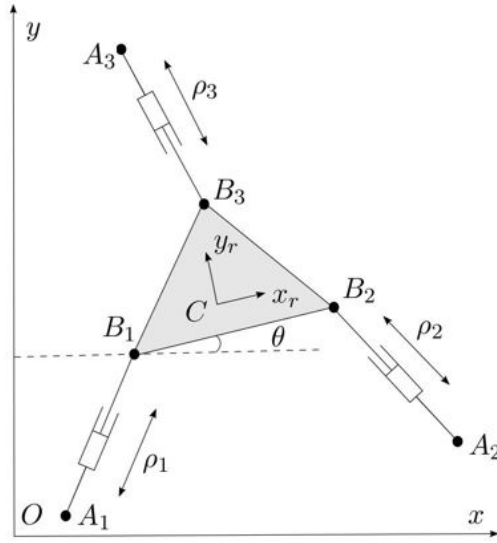
Présentons tout d'abord un algorithme qui permet le calcul de l'espace de travail statique pour une orientation donnée et une charge connue appliquée à la plate-forme. Nous allons illustrer cet algorithme sur le robot à déplacement dans le plan, 3-RPR [29],[40]. Ce robot figure parmi les architectures parallèles les plus étudiées car son "alter ego" spatial n'est autre que la plate-forme de Gough [16], [17], [22]. De plus, outre sa relative simplicité, de nombreux travaux sur ses singularités ont déjà été effectués. Ainsi nous connaissons les lieux singuliers du 3-RPR et des travaux sur les moyens de contourner les singularités pour changer de mode d'assemblage ont été développés [53].

3.2 Définition de l'espace de travail statique

Nous avons choisi d'illustrer l'espace de travail statique pour le robot plan 3-RPR (figure 3.1) mais la définition reste valable pour d'autres types de robots parallèles pouvant compter jusqu'à 6 degrés de liberté.

On rappelle que la notation A_i désigne les points d'attache des jambes attachés avec la base et B_i les points d'attache du mécanisme situés sur la plate-forme mobile. Leurs coordonnées sont respectivement exprimées dans le repère de référence $\Omega_O(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$ et le repère mobile $\Omega_C(C, \mathbf{x}_r, \mathbf{y}_r, \mathbf{z}_r)$. La longueur des jambes du robot est notée $\rho_i = \|\mathbf{A}_i \mathbf{B}_i\|$. Enfin, nous désignons par \mathcal{F} l'effort extérieur exercé par les jambes sur la plate-forme mobile et τ représente les forces dans les jambes. La relation fondamentale de la statique, qui consiste à établir l'équilibre mécanique du robot, nous fournit une relation entre τ et \mathcal{F} .

$$J^{-T} \tau = \mathcal{F} \quad (3.3)$$

FIG. 3.1: Le robot 3 – RPR

avec J^{-T} la transposée de la matrice jacobienne cinématique inverse minimale J^{-1} dont une ligne s'écrit :

$$(\mathbf{J}^{-1})_i = \left(\frac{\mathbf{A}_i \mathbf{B}_i}{\rho_i} \quad \frac{\mathbf{CB}_i \times \mathbf{A}_i \mathbf{B}_i}{\rho_i} \right) \quad (3.4)$$

Nous déduisons de J^{-1} la matrice M dont une ligne \mathbf{M}_i est donnée par :

$$\mathbf{M}_i = (\mathbf{A}_i \mathbf{B}_i, \mathbf{CB}_i \times \mathbf{A}_i \mathbf{B}_i)$$

Si le rang de la matrice J^{-T} est inférieur ou égal à 3 dans notre cas, on se trouve en singularité. Comme le déterminant est invariant par transposition, nous pourrions également considérer la jacobienne cinématique inverse. Si elle est singulière alors la matrice M l'est aussi et réciproquement. Nous définissons les matrices N_i en substituant la i -ème colonne de J^{-T} par \mathcal{F} . En utilisant la règle de Cramer, nous pouvons exprimer les composantes τ_i en fonction de M, N_i et ρ_i :

$$\tau_i = \rho_i \frac{|N_i|}{|M|} \quad \text{si } |M| \neq 0 \quad (3.5)$$

Si M est singulière le système (3.3) est sous-déterminé mais pourra tout de même admettre des solutions admissibles, c'est-à-dire ne conduisant pas à une détérioration du mécanisme. Aussi il apparaît que statiquement parlant, déterminer si J^{-T} est singulière ou non n'est pas suffisant. Nous pouvons effectivement nous trouver sur une position singulière X_s si la trajectoire suivie conduit à des déterminants des N_i qui tendent vers zéro quand on se rapproche de X_s .

Cependant à l'approche d'une singularité, les forces subies par les jambes peuvent se révéler très importantes et conduire à la détérioration du robot. Il est légitime de supposer la connaissance d'une force de seuil τ_{max} pour chaque jambe. Par exemple on

peut définir τ_{max} comme la force maximale que peut supporter l'élément mécanique le plus faible de la jambe. Sans perte de généralité, nous supposons ici que cette force de seuil est identique pour les trois jambes. Nous définissons alors *l'espace de travail statique* W_s .

Définition : Soit χ une pose de l'organe terminal et W l'espace de travail du robot tel que l'impose les contraintes géométriques. Nous définissons *l'espace de travail statique* W_s pour une charge donnée \mathcal{F} par :

$$W_s = \{\chi \in W \text{ telle que } \forall i \ |\tau_i| \leq \tau_{max}\}$$

L'espace de travail statique est donc l'ensemble des poses qui préservent l'intégrité du robot. W_s est un élément d'un espace à n dimensions, où n représente le nombre de paramètres nécessaires à la représentation d'une pose. Il n'est donc graphiquement pas représentable si $n > 3$.

3.2.1 Espace de travail statique du 3-RPR

Nous nous intéressons ici au calcul de l'espace de travail statique *pour une charge fixée*. L'espace de travail statique est à priori une variété immergée dans un espace de dimension 3. Nous allons ici simplifier le problème en supposant que *l'orientation est constante*. Nous allons alors montrer que l'on peut calculer la frontière de W_s .

Dans ce but, on va considérer le robot planaire 3-RPR (figure 3.1), où la position \mathbf{X} de la plate-forme mobile peut être paramétrée par les coordonnées (x, y) d'un point C appartenant à la plate-forme. Pour paramétrer l'orientation θ , on considère θ comme l'angle entre les axes \mathbf{x} et \mathbf{x}_r des deux repères Ω_O et Ω_C .

Remarquons que seul 2 paramètres varient (ceux qui déterminent la position du point C), nous pouvons alors considérer τ comme une fonction de \mathbf{X} : $\tau(\mathbf{X})$. Notons également que les longueurs des jambes ρ_i sont données par :

$$\rho_i = \sqrt{(x - U_i)^2 + (y - V_i)^2} \quad (3.6)$$

avec U_i et V_i des constantes qui dépendent uniquement de la géométrie du robot, de la jambe i et de l'orientation θ .

3.3 Frontière de l'espace de travail statique

Nous nous intéressons ici aux poses qui appartiennent à la frontière de l'espace de travail statique.

Proposition A : Une pose χ_0 appartient à la frontière de l'espace de travail statique si et seulement si les deux conditions suivantes sont vérifiées :

- *Condition 1* : Pour au moins une jambe i du robot on vérifie $|\tau_i| = \tau_{max}$ et pour les autres jambes la contraintes $|\tau_j| \leq \tau_{max}$ est vérifiée.

• *Condition 2* : Il existe au moins un déplacement infinitésimal qui conduit à $|\tau_i| > \tau_{max}$.

Preuve : La seconde condition est triviale pour la définition d'une frontière. Il reste donc à prouver la condition 1.

Par définition les forces τ_j des jambes, pour les poses de l'espace de travail statique vérifient $|\tau_j| \leq \tau_{max}$. Il reste donc à prouver que l'une d'entre elles au moins vérifie $|\tau_j| = \tau_{max}$.

Soit une pose χ_0 telle que toutes les forces articulaires vérifient $|\tau_k| < \tau_{max}$. Supposons que cette pose appartienne à la frontière et montrons par l'absurde que ceci est impossible car il n'existe pas de déplacement infinitésimal pour une jambe i conduisant à $|\tau_i| > \tau_{max}$.

Soit B une boule de centre χ_0 et de rayon r et soit χ_f une pose appartenant à cette boule. On désigne par J_j^T la j -ème ligne de la matrice J^T (matrice inverse de J^{-T} que nous ne devons pas confondre avec la transposée de la jacobienne cinématique inverse). D'après le théorème des fonctions implicites, les forces articulaires en χ_f sont obtenues par :

$$\tau_j(\chi_f) = \left(J_j^T(\chi_0) + \frac{\partial J_j^T(\chi_1)}{\partial \chi} (\chi_f - \chi_0) \right) \mathcal{F} \quad (3.7)$$

où χ_1 est une pose appartenant à la boule B . A l'intérieur de B la quantité $\frac{\partial J_j^T(\chi_1)}{\partial \chi}$ est bornée et donc nous pouvons définir une valeur $U > 0$ telle que :

$$-r_1 U \leq \frac{\partial J_j^T(\chi_1)}{\partial \chi} \mathcal{F} \leq r_1 U \quad \text{avec } r_1 = \|\chi_f - \chi_0\| \quad (3.8)$$

On peut donc écrire :

$$|\tau_j(\chi_f)| \leq |J_j^T(\chi_0) \mathcal{F}| + U \|\chi_f - \chi_0\| \quad (3.9)$$

Par définition $|J_j^T(\chi_0) \mathcal{F}| < \tau_{max}$ et il existe donc un $\epsilon > 0$ tel que

$$|J_j^T(\chi_0) \mathcal{F}| = \tau_{max} - \epsilon \quad (3.10)$$

On a donc :

$$|\tau_j(\chi_f)| \leq \tau_{max} - \epsilon + U \|\chi_f - \chi_0\| \quad (3.11)$$

En conséquent pour tous les χ_f de B vérifiant $\|\chi_f - \chi_0\| < \frac{\epsilon}{U}$, nous avons $|\tau_j| < \tau_{max}$ ce qui contredit la condition 2 : donc χ_0 ne peut pas appartenir à la frontière. \square

Soit Γ_s la frontière de l'espace de travail statique. Si nous sommes capable de calculer cette frontière alors nous verrons qu'il est possible d'obtenir W_s dans sa globalité. Nous allons donc dans le prochain paragraphe nous intéresser à la caractérisation de Γ_s .

3.4 Éléments de la frontière

Γ_s va être composé de portions de courbes pour lesquelles $\tau_i = \tau_{max}$ ou $\tau_i = -\tau_{max}$. Comme nous avons $\tau_i = \rho_i \frac{|N_i|}{|M|}$ si $|M| \neq 0$ nous pouvons considérer deux types de courbes potentiellement constitutives de Γ_s :

1) Les courbes planes du *type n* correspondent au choix $\tau_i = \tau_{max}$ avec $|M| \neq 0$:

$$\tau_{max}|M| - \rho_i|N_i| = 0 \quad (3.12)$$

2) Les courbes planes du *type m* correspondent au choix $\tau_i = -\tau_{max}$ pour $|M| \neq 0$:

$$\tau_{max}|M| + \rho_i|N_i| = 0 \quad (3.13)$$

Le 3 – RPR ayant trois jambes, nous avons donc trois courbes de *type n* et trois de *type m*. Remarquons cependant que les points des courbes de *type n* et *m* ne sont pas forcément tous sur la frontière Γ_s , puisqu'ils doivent aussi vérifier les autres conditions de la proposition A. Les courbes (3.12) et (3.13) ne sont pas algébriques en raison de la présence d'une racine carré dans le terme ρ_i (voir l'équation 3.6). La manipulation de ce type de non-linéarité étant délicate, nous élevons au carré ces termes en multipliant les équations (3.12) et (3.13) entre elles, ce qui conduit à :

$$\tau_{max}^2|M|^2 - \rho_i^2|N_i|^2 = 0 \quad (3.14)$$

Ces équations maintenant algébriques caractérisent simultanément les courbes de *type n* et *m* et nous les désignons comme des courbes de *type q*. Dans le cas du 3 – RPR, les éléments de M et N_i étant linéaire en x, y le degré total de $|M|$ en x, y est 3 et de 2 pour $|N_i|$. Comme ρ_i^2 est de degré 2 en x, y , nous pouvons qualifier les équations (3.14) de *sextiques*, c'est-à-dire de degré total 6 en x, y .

L'établissement des courbes de *type n* (3.12) et de *type m* (3.13) n'est valable que dans le cas où J^{-T} est régulière, c'est-à-dire pour $|M| \neq 0$, et/ou $\rho_i \neq 0$ ce qui correspond au cas où J^{-1} n'est pas définie. Si nous supposons que $|M| = 0$ (le robot est en configuration singulière), nous obtenons une valeur finie pour τ_i dans seulement deux cas :

- Cas 1 : $|M| = 0$ et $|N_i| = 0$
- Cas 2 : $|M| = 0$ et $\rho_i = 0$

On appelle les poses vérifiant à la fois les équations $|M| = 0$ et $|N_i| = 0$ les *points N* et celles vérifiant $|M| = 0$ et $\rho_i = 0$ les *points R*. Remarquons que les *points R* et les *points N* appartiennent à la fois aux courbes de *type n* et de *type m*.

Nous allons maintenant déterminer ces différentes courbes et points puis montrer comment à partir de ces éléments nous pouvons déterminer la frontière Γ_S de l'espace de travail statique W_S .

3.5 Calcul de la frontière

Dans un premier temps, nous allons établir un certain nombre de définitions indispensables à la compréhension de l'algorithme calculant la frontière de l'espace de travail statique.

3.5.1 Les points clés

Définition : les *points clés* sont définis comme :

- des intersections d'une courbe de *type* n ou m avec toutes les autres courbes de type n et m .
- des points N et R .

Les *points clés* jouent un rôle déterminant dans le calcul de la frontière Γ_S de l'espace de travail statique en raison du théorème suivant :

Théorème : Si W_s n'est pas vide sa frontière Γ_s est constituée :

- *Cas 1* : soit par la frontière de l'espace de travail W .
- *Cas 2* : soit par des arcs de courbes de type n ou de type m dont les extrémités sont des points clés.
- *Cas 3* : soit par une courbe de type n ou m entière, ou une portion fermée d'une telle courbe.

Démonstration :

• *Cas 1* : La relation fondamentale de la statique (3.3) implique $|\tau| \leq |J^{-T}| |\mathcal{F}|$. Nous pouvons affirmer que $|J^{-T}| > 0$ est bornée sur l'espace de travail W par $U = \text{Max}(|J^{-T}|) > 0$ et $V = \text{Min}(|J^{-T}|) > 0$ donc nous pouvons écrire :

$$V|\mathcal{F}| \leq |\tau| \leq U|\mathcal{F}| \quad (3.15)$$

Si pour toute pose de l'espace de travail nous vérifions $U|\mathcal{F}| < \tau_{max}$ alors l'espace de travail est inclus dans l'espace de travail statique.

De la même manière si nous vérifions $V|\mathcal{F}| > \tau_{max}$ pour toute pose de l'espace de travail alors l'espace de travail statique est vide.

• *Cas 2* : Soit une portion de courbe de *type* n ou m entre deux points clés successifs K_i et K_j . On va montrer que si un point P quelconque de la portion de courbe délimitée par K_i et K_j fait partie de la frontière Γ_s de l'espace de travail statique alors tous les points de la portion de courbe vont faire partie de Γ_s .

En effet, si un point P n'appartient pas à la liste des points clés, mais qu'il fait partie de la frontière Γ_S alors on peut affirmer qu'il vérifie $|\tau_i| = \tau_{max}$ mais aussi $\forall j \neq i$ $|\tau_j| < \tau_{max}$ (d'après la définition de Γ_s et des points clés).

Supposons alors qu'il existe sur la portion de courbe étudiée un point P_0 qui n'appartient pas à la frontière de l'espace de travail statique. Ce point devra forcément vérifier qu'il existe un j_0 tel que $|\tau_{j_0}| > \tau_{max}$. La continuité de $|\tau|$ implique donc qu'il existe donc un autre point K_0 telle que $|\tau_i| = \tau_{max}$, $|\tau_{j_0}| = \tau_{max}$ et $\forall j \neq i, j_0$ $|\tau_j| < \tau_{max}$. Le point K_0 serait donc un point clé (car intersection de deux courbes de *type* n ou m). Par conséquent, nous contredisons le fait que la portion de courbe soit délimitée par deux points clés successifs. Ainsi la portion de courbe entière délimite une partie de Γ_s .

• *Cas 3* : Soit une portion de courbe fermée de type n ou m n'admettant aucune intersection avec les autres courbes. Par définition sur l'ensemble de la portion les points vérifient $|\tau_i| = \tau_{max}$. Si nous supposons de plus que $\forall j \neq i \quad |\tau_j| < \tau_{max}$ alors cette portion de courbe appartient à la frontière γ_s si la condition 2 de la proposition A est vérifiée (il existe un déplacement infinitésimal conduisant à $|\tau_i| > \tau_{max}$) pour tous les points constitutifs de cette portion de courbe. \square

3.5.2 Détermination des points clés

Rappelons que l'on considère pour l'instant les points clés comme les intersections de toutes les courbes qu'elles soient du type n ou m , mais aussi les points R et N.

Afin de calculer les intersections entre les courbes de type n ou m , nous utilisons les courbes du type q que nous avons définies précédemment. Il est clair que les intersections entre 2 courbes de type $q : (q_i, q_j)$ donnent tous les points clés qui résultent des intersections des paires de courbes n_i, n_j, m_i, m_j à savoir $(n_i, m_j), (n_j, m_i), (n_i, n_j)$ et (m_i, m_j) .

Afin d'obtenir les points clés résultant de l'intersection des courbes de type n ou m calculons d'abord le résultant des équations de type q correspondantes, par exemple en x . Nous obtenons donc un polynôme univarié en y . Résolvons alors numériquement ce polynôme et nous obtenons donc les composantes y des solutions. Après détermination des solutions en y , on remonte à la valeur de la composante en x en résolvant les équations initiales, qui sont devenues univariées en x . Cela nous permet d'éliminer les racines en y incorrectes qui sont parfois générées par l'utilisation du résultant.

Nous allons obtenir les points R et N en utilisant le même principe. Ainsi pour les points R , on utilisera d'abord le résultant entre les polynômes d'équation $\rho_i^2 = 0$ et $|M| = 0$ tandis que les points N seront obtenus à partir du résultant de $|M| = 0$ et $|N_i| = 0$.

3.5.3 Les points clés auxiliaires

La détermination des points clés ne permet pas de traiter le cas 3 (portion fermée d'une courbe de type n ou m). De plus, il peut se poser le cas où une courbe part à l'infini tout en faisant partie de Γ_s . Dans ces deux cas nous ne pourrions pas appliquer notre algorithme soit parce que nous n'avons pas de point clé, soit parce qu'ils sont en nombre impair. Pour résoudre ce problème nous allons introduire des *points clés auxiliaires*.

Remarquons tout d'abord que l'espace de travail statique W_s qui nous intéresse est sa partie incluse dans l'espace de travail W . Les contraintes conduisant à limiter W sont très diverses. Pour en simplifier le calcul nous allons supposer que nous pouvons définir une boîte $\mathcal{B} = [x_{min}, x_{max}] \times [x_{min}, x_{max}]$ qui englobe W . On définit comme point clé auxiliaire les intersections des courbes q avec \mathcal{B} . De plus on élimine les points clé non contenus dans \mathcal{B} .

Nous pouvons facilement déterminer ce type de points clés auxiliaire en remplaçant la variable x ou y suivant le cas soit par x_{min} ou x_{max} , soit par y_{min} ou y_{max} , puis résoudre le polynôme univarié qui correspond à l'équation de type q . Enfin il ne nous reste plus

qu'à attribuer à la bonne courbe les solutions obtenues.

Le cas 3 où soit la courbe de type n ou m ou une partie fermée de cette courbe définit une partie de Γ_s se montre plus problématique. Afin de le résoudre, nous définissons comme point clé auxiliaire les points où la tangente à la courbe est soit horizontale, soit verticale. Ils peuvent être calculés à partir du résultant de q_i et $\frac{\partial q_i}{\partial x}$ pour une tangente verticale et de la même manière q_i et $\frac{\partial q_i}{\partial y}$ pour les tangentes horizontales, courbes qui sont toutes deux des sextiques en x, y .

À l'issue de ces étapes, nous disposons d'une liste de points clés et une liste de points clés auxiliaire. Si nous ne faisons plus de différences entre les points clés et les points clés auxiliaires, nous pouvons généraliser le cas 2 du théorème et affirmer que Γ_s est constituée d'arcs de courbes de type n ou m compris entre deux points clés successifs.

3.5.4 Appartenance d'une pose à la frontière

Nous venons de voir, qu'avec l'assimilation des points clés auxiliaires comme points clés à part entière, la frontière de l'espace de travail statique est constituée par des arcs de courbes de type n ou m délimités par deux points clés successifs. Nous avons établi les conditions pour lesquelles un point appartient à la frontière. Montrons comment prouver qu'un point appartenant à une courbe n ou m se situe sur la frontière.

Si un point $P = (x_p, y_p)$ appartient à une courbe de type n ou m , nous pouvons affirmer qu'il existe une des composantes τ_i telle que $|\tau_i| = \tau_{max}$. Cependant nous ne pouvons pas établir de condition supplémentaire sur les autres composantes des forces articulaires $\forall j \neq i \tau_j$. Nous avons identifié deux cas :

Cas 1 : Si $\exists j \neq i \tau_j > \tau_{max}$ alors nous pouvons immédiatement conclure que le point P n'appartient pas à la frontière, puisque ce dernier se trouve dans la zone où l'intégrité du robot n'est pas préservée.

Cas 2 : Si $\forall j \neq i \tau_j < \tau_{max}$, il nous reste à vérifier la condition 2 de la proposition A, c'est-à-dire qu'il existe un déplacement infinitésimal conduisant à $|\tau_i| > \tau_{max}$. Supposons que l'on traite la courbe n_i , cela implique que pour tous les points de la courbe on a $\tau_i = \tau_{max}$. Rappelons que τ_i peut s'écrire sous la forme d'une fonction en x, y : $\tau_i = f(x, y)$. Supposons que la fonction f est différentiable en P . Si pour le point P on a $\frac{\delta f}{\delta x}(x_p, y_p) \neq 0$ alors il existe un déplacement infinitésimal selon la direction x conduisant à $\tau_i > \tau_{max}$:

- dans le sens des abscisses croissantes pour le cas où $\frac{\delta f}{\delta x}(x_p, y_p) > 0$
- dans le sens des abscisses décroissantes pour le cas où $\frac{\delta f}{\delta x}(x_p, y_p) < 0$

De la même manière si $\frac{\delta f}{\delta y}(x_p, y_p) \neq 0$ alors on peut affirmer qu'il existe un déplacement infinitésimal selon la direction y conduisant à $\tau_i > \tau_{max}$. On pourra donc affirmer que pour un point P appartenant à une courbe de type n ou m , il existe un déplacement

infinitésimal conduisant à $|\tau_i| > \tau_{max}$ à partir du moment où l'on vérifie : $\frac{\delta f}{\delta x}(x_p, y_p) \neq 0$
ou $\frac{\delta f}{\delta y}(x_p, y_p) \neq 0$.

3.5.5 Principe théorique du calcul de la frontière

Les étapes précédentes nous ont permis d'attribuer à chaque courbe une liste non ordonnée de points clés. Nous savons également déterminer si un point d'une courbe n ou m situé entre deux points clés successifs fait partie ou non de la frontière. Si un point P non clé d'un arc appartient à la frontière, alors l'arc entier est une partie constitutive de Γ_s . Nous en déduisons un algorithme de base :

Etape 1 : Calcul des points clés (rappelons que nous ne faisons plus de différences entre les points clés des points clés auxiliaires).

Etape 2 : Ordonner les points clés pour chaque courbe, c'est-à-dire déterminer quels sont les points clés successifs, ce qui nous permettra d'identifier les différents arcs de courbes qui peuvent appartenir à Γ_s .

Etape 3 : Pour chaque arc de courbe (portion d'une courbe entre deux points clés successifs), prendre un point quelconque P sur l'arc et vérifier s'il appartient à la frontière ou non : s'il appartient à la frontière, alors l'arc de la courbe entre les deux points clés fait partie de Γ_s . Le résultat final de l'algorithme sera la liste de tels arcs.

Nous ne sommes malheureusement pas parvenu à obtenir une représentation paramétrique des courbes de type n ou m . En conséquence l'étape d'ordonnement des points clés devient un obstacle majeur puisque nous ne sommes pas en mesure de déterminer parmi la liste des points clés lesquels sont des points clés successifs. Confrontés à ce problème, nous avons adapté notre méthode pour déterminer une approximation polygonale de la frontière de l'espace de travail statique.

3.5.6 Mise en pratique

Nous cherchons à obtenir une approximation polygonale de Γ_s . Nous disposons d'une liste de points clés non ordonnée compris dans une boîte \mathcal{B} qui englobe l'espace de travail. La première étape pour le traitement des points clés consiste à éliminer ceux pour lesquels il existe une composante τ_i telle que $|\tau_i| > \tau_{max}$ car il ne peuvent pas appartenir à la frontière. Nous ordonnons ensuite de manière croissante les points clés selon leur abscisse mais aussi leur ordonnée. Nous disposons donc pour chaque courbe de deux listes de points clés : L_x (abscisse croissante) et L_y (ordonnée croissante).

Nous allons maintenant décrire un processus permettant d'obtenir un arc de la frontière entre deux points clé successifs d'une courbe C de type n ou m . Partons d'un point clé K_p de coordonnées (x_p, y_p) et calculons la tangente $T = (t_x, t_y)$ à la courbe traitée en K_p . Distinguons deux cas :

- *Cas 1* : $t_x^2 > t_y^2$ qui correspond à une courbe plutôt "horizontale"
- *Cas 2* : $t_x^2 \leq t_y^2$ qui correspond à une courbe plutôt "verticale"

Nous allons détailler le processus en supposant que K_p vérifie $t_x^2 > t_y^2$ (figure 3.2) mais nous comprendrons qu'une méthode analogue existe si $t_x^2 \leq t_y^2$ (ce cas sera traité ultérieurement). Notre objectif consiste dans un premier temps à trouver un point M_1 à "droite" de K_p , au plus à une distance μ (on expliquera pourquoi ultérieurement) et qui soit sur la même branche de la courbe C que le point K_p . Comme nous cherchons à déterminer la frontière le point M_1 doit appartenir à Γ_S .

Etape 1 : Trouver un point M_1 à "droite" de K_p : Pour être certain que M_1 se situe bien à droite, nous fixons son abscisse telle que $x_{M_1} = x_p + \epsilon$, avec $\epsilon > 0$ arbitrairement petit (figure 3.3). Il nous reste alors à calculer l'ordonnée y_{M_1} de M_1 et vérifier qu'avec K_p ils appartiennent à la même branche de la courbe C .

Etape 2 : Montrer que $\|M_1 K_p\| \leq \mu$ et que M_1 et K_p appartiennent à la même branche de la courbe C *Moyen* : nous allons avoir besoin du *théorème de Kantorovitch*.

Théorème de Kantorovitch 1D :

Soit une équation du type $f(x) = 0$ où f est une fonction de classe C^2 sur U .

Soit x_0 un point et $U = \{x : \|x - x_0\| \leq 2B_0\}$

Si l'on vérifie les conditions suivantes :

- $f'(x_0) \neq 0$ et $|\frac{1}{f'(x_0)}| = A_0$
- $|\frac{f(x_0)}{f'(x_0)}| = B_0$
- la dérivée seconde est bornée sur U : $\forall x \in U \ |f''(x)| < C$
- les constantes A_0 , B_0 et C vérifient $2A_0B_0C \leq 1$

Alors il y a une unique solution à l'équation $f(x) = 0$ dans U et la méthode de Newton converge avec x_0 comme première estimation.

Nous injectons x_{M_1} dans l'équation de la courbe C qui s'écrit sous la forme $f(x, y) = 0$.

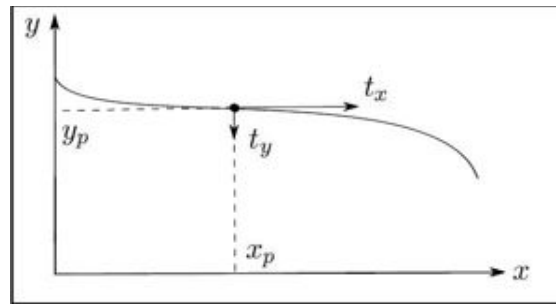


FIG. 3.2: Cas où les composantes de la tangente au point K_p sont telles que $t_x^2 > t_y^2$

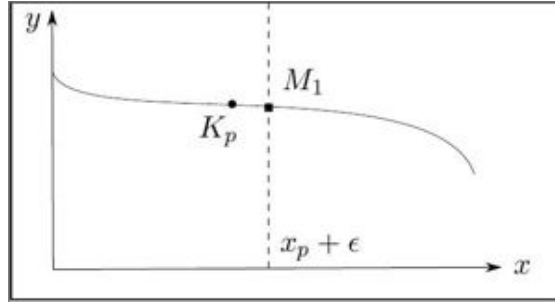


FIG. 3.3: On cherche à déterminer M_1 à droite de K_p , on fixe donc $x_{M_1} = x_p + \epsilon$

Nous obtenons une équation non linéaire à une seule inconnue $f(x_{M_1}, y) = g(y) = 0$. Nous appliquons le théorème de Kantorovitch pour déterminer s'il existe une solution unique y_{M_1} à l'équation $g(y) = 0$ dans la boule centrée en K_p et de rayon μ (figure 3.4). L'unicité de la solution implique que M_1 et K_p appartiennent à la même branche de la courbe C .

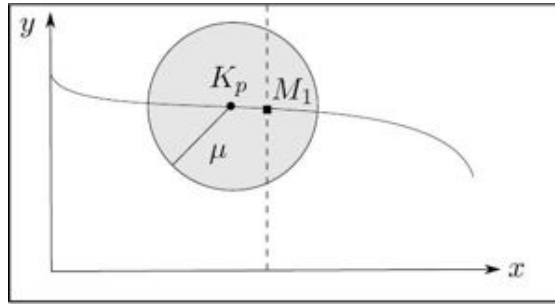


FIG. 3.4: On applique le théorème de Kantorovitch sur la boule centrée en K_p et de rayon μ , pour déterminer l'ordonnée du point M_1

- Si les conditions du théorème de Kantorovitch sont vérifiées, la méthode de Newton converge avec y_p comme estimation initiale. Nous avons entièrement déterminé le point $M_1(x_{M_1}, y_{M_1})$ qui se situe sur la même branche de la courbe C que le point clé de départ K_p et vérifie la condition $\|\mathbf{M}_1 \mathbf{K}_p\| < \mu$. On comprendra que μ doit être inférieur à $2B_0$.
- Sinon divisons ϵ par 2 jusqu'à obtenir la vérification du théorème de Kantorovitch, vers une unique solution.

Etape 3 : Détermination d'un arc de Γ_s délimité par K_p à "gauche" et un autre point clé à "droite".

Vérifions ensuite que M_1 est bien sur la frontière. En effet si ce n'est pas le cas, nous pouvons conclure qu'il n'existe pas de branche de la courbe C partant de K_p vers la "droite" et qui soit un élément de Γ_s . Pour continuer la description de notre algorithme, nous supposons que le point M_1 est bien sur Γ_s . Dans ce cas le segment $K_p M_1$ constitue une approximation polygonale d'une "petite portion" de la frontière (figure 3.5). Pour obtenir un arc, délimité par deux points clés, nous réitérons le processus en

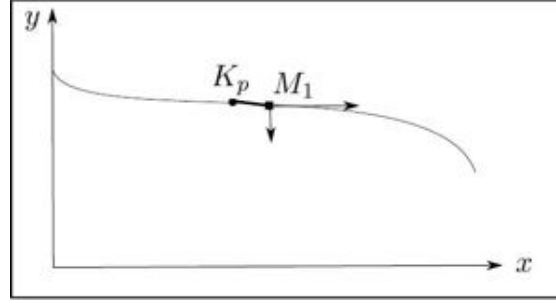


FIG. 3.5: Si les conditions de Kantorovitch sont vérifiées, nous avons déterminé un point M_1 . Vérifions qu'il appartient à Γ_s . Si c'est le cas alors nous avons déterminé une petite portion de Γ_s , continuons le processus en cherchant un point "à droite" de M_1

partant cette fois ci de M_1 au lieu de K_p .

Arrêtons ce processus à la i -ème itération si nous rencontrons l'un des cas suivants :

a) le point M_i est l'un des points clés de la courbe C . Nous ajusterons au besoin $\epsilon = \epsilon_{ajuste}$ au cours du processus quand M_{i-1} est "proche" de K_{p+1} afin d'obtenir $x_{M_i} = x_{p+1} = x_{M_{i-1}} + \epsilon_{ajuste}$ (figure 3.6). Il nous suffit d'utiliser la liste L_x pour cela. Si nous vérifions que le point M_i et le point K_{p+1} sont identique alors nous avons déterminé une approximation polygonale de l'arc de la courbe C délimité par K_p et K_{p+1} . Si au contraire $y_{M_i} \neq y_{p+1}$, nous chercherons à rejoindre K_{p+2} et ainsi de suite jusqu'à rejoindre un autre point clé ou rencontrer le cas b).

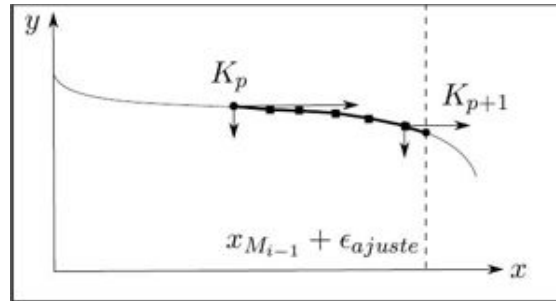


FIG. 3.6: On arrête le processus si on rejoint un autre point clé (cas a)

b) $t_x^2 \leq t_y^2(x_M, y_m)$ (figure 3.7) : Nous traiterons ce cas ultérieurement car il correspond au cas 2 précédemment cité. Le point M_i sera stocké comme un point clé dans les listes L_x et L_y .

Ainsi nous pouvons considérer qu'à la fin de cette étape, nous avons déterminé une approximation polygonale de l'arc de courbe C délimité par K_p "à gauche" et un autre point clé "à droite". Rappelons que nous ne déterminerons cet arc de courbe que s'il fait partie de Γ_s . Il nous reste à suivre le même processus afin d'obtenir une approximation polygonale de l'arc de courbe C délimité par K_p "à droite" et un autre point clé "à gauche". Pour ce cas, nous utiliserons un $\epsilon < 0$ arbitrairement petit pour incrémenter l'abscisse : $x_{M_1} = x_p + \epsilon$.

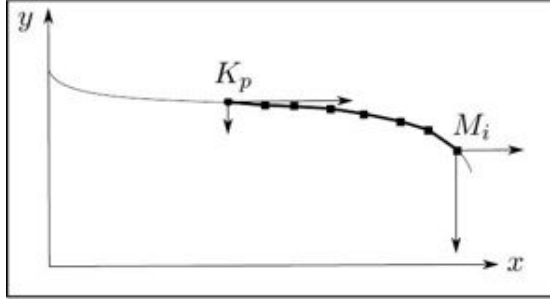


FIG. 3.7: On arrête le processus si la relation d'ordre des tangentes change (cas b)

Si le point clé K_p vérifie $t_x^2 \leq t_y^2(x_M, y_m)$, nous cherchons l'approximation polygonale des arcs de courbe C délimités par K_p "en haut" ou "en bas" et rejoignant un autre point clé. Nous utilisons un processus analogue mais nous incrémentons non pas l'abscisse d'un paramètre ϵ mais l'ordonnée : $y_{M_1} = y_p + \epsilon$. Nous utiliserons alors la liste L_y pour vérifier que nous nous situons "près" d'un autre point clé.

Si ce processus est appliqué pour tous les points clés et toutes les courbes de type n ou m nous obtenons une approximation polygonale d'arcs de courbes qui sont des éléments de la frontière, l'union de ces arcs étant une approximation polygonale de Γ_s .

L'algorithme se termine lorsque l'ensemble des points clés de la liste a été marqué. Chaque point clé est marqué lorsqu'il sert de point de départ ou qu'il est "découvert" lors du processus. Le marquage inclut la direction de l'approximation polygonale qui a conduit au point clé et un point clé a été totalement exploité s'il est marqué selon deux directions (à l'exception des points-clés sur la frontière de la boîte englobante qui ne sont marqués que dans une direction).

3.5.7 Variation de l'algorithme

Nous pouvons adapter cet algorithme pour prendre en compte des contraintes supplémentaires sur les mouvements du robot. Par exemple, on peut limiter les évolutions du robot en imposant que les ρ_i restent dans un intervalle $[\rho_{min}, \rho_{max}]$. Suivant que l'on étudie la contrainte $\rho_i = \rho_{min}$ ou $\rho_i = \rho_{max}$, les points B_i ne peuvent appartenir qu'au cercle de centre A_i et de rayon ρ_{min} ou ρ_{max} . Comme l'orientation est constante, cette contrainte sur les B_i impose une contrainte similaire sur C , qui doit appartenir à un cercle de même rayon mais dont le centre s'obtient à partir des centres de cercle pour les B_i par translation de vecteur $B_i C$. Nous calculons alors les intersections des cercles avec les courbes de type n et m et nous les stockons comme des points clés. Comme nous connaissons la représentation paramétrique d'un cercle, il est possible d'ordonner les points clés des cercles contrairement à ce que nous avons pu faire pour les courbes de type n ou m . Ces points clés délimitent des arcs de cercle pour lesquels nous devons vérifier qu'ils sont des éléments de la frontière ou non. Pour cela il suffit de vérifier qu'un point quelconque de cet arc se situe bien sur Γ_s .

3.6 Résultats

3.6.1 Exemple d'espace de travail statique

L'algorithme a été développé sous *Maple*. Les paramètres géométriques du mécanisme considéré sont donnés par :

$$\begin{aligned} OA_1 &= (0, 0) & CB_{r1} &= (-4, 4) \\ OA_2 &= (20, 0) & CB_{r2} &= (4, -4) \\ OA_3 &= (12, 10) & CB_{r3} &= (0, 2) \end{aligned}$$

Nous avons utilisé un incrément $\epsilon = 0.03$ pour toutes les courbes. Le tableau 3.1 donne les valeurs de l'orientation, de la charge appliquée à la plate-forme ainsi que la valeur de la force de τ_{max} pour chaque courbe. La figure 3.8 illustre l'ensemble des courbes de type

figure	\mathcal{F}	τ_{max}	θ (exprimé en radians)
3.8	(1, 1, 1)	5	-0.1
3.9	(1, 1, 1)	5	-0.1
3.10	(1, 1, 1)	3	-0.1
3.11	(1, 1, 1)	2	-0.1
3.12	(1, 1, 1)	1.5	-0.1
3.13	(1, 1, 1)	1.3	-0.1
3.14	(4, 0, 0)	30	0
3.15	(4, 0, 0)	30	0

TAB. 3.1: Paramètres utilisés pour tracer les différentes courbes

n ou m impliquées dans un calcul de l'espace de travail statique, la figure 3.9 présentant l'espace de travail statique résultant. Nous y voyons l'espace de travail statique qui est constitué de trois régions, connectées les unes aux autres par différents points singuliers. Ainsi, nous voyons bien que la seule recherche des singularités pour un mécanisme n'est pas suffisante.

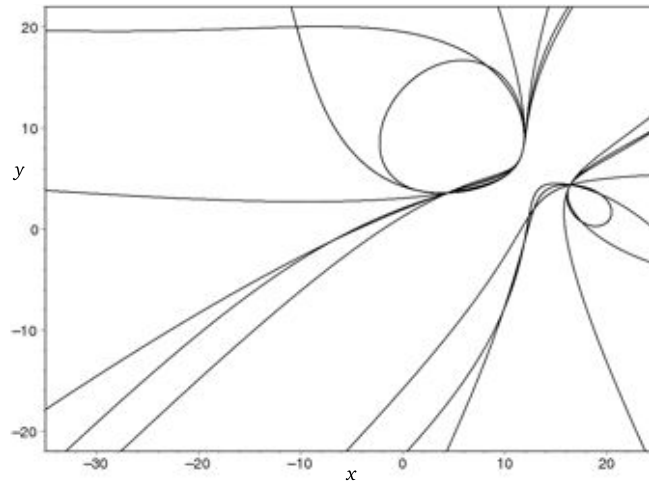


FIG. 3.8: Les courbes nécessaires au calcul de W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 5$, $\theta = -0.1$

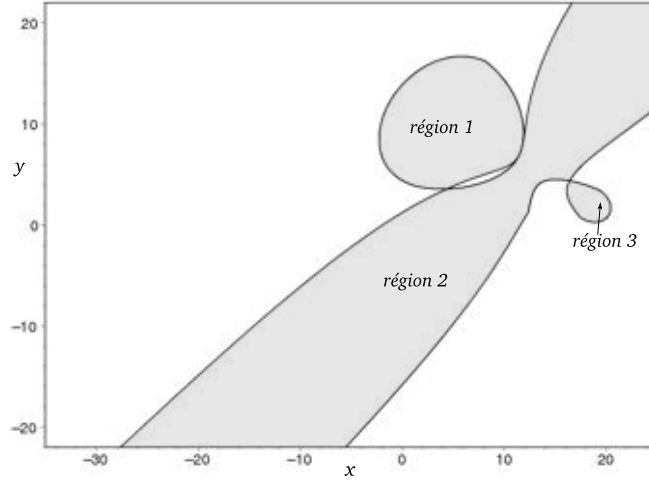


FIG. 3.9: En gris W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 5$ et $\theta = -0.1$

Intéressons nous par la suite aux coupes planes de l'espace de travail statique visibles sur les figures 3.10, 3.11, 3.12 et 3.13. Pour tracer ces frontières, nous avons conservé la même charge appliquée au mobile et la même orientation de la plate-forme que sur la figure 3.9. mais nous faisons varier la valeur du seuil τ_{max} . Bien évidemment, nous

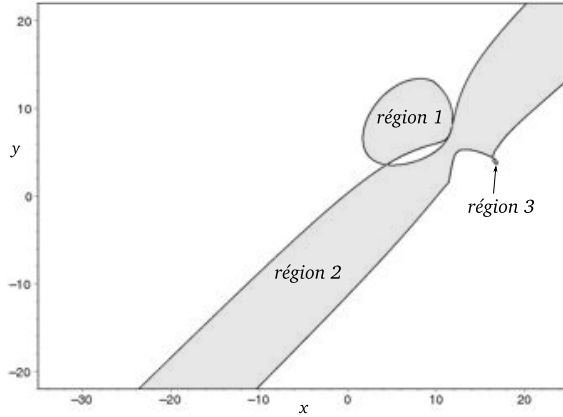


FIG. 3.10: En gris W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 3$ et $\theta = -0.1$ rad

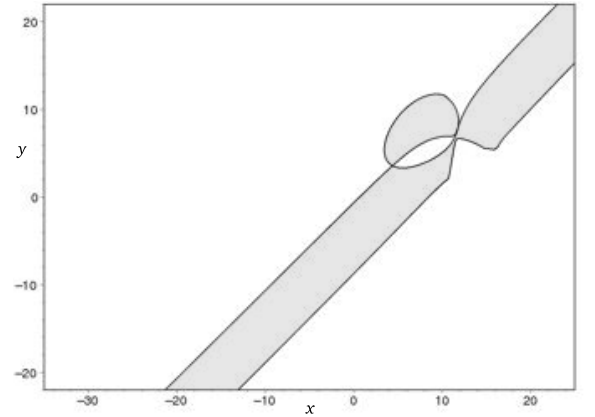


FIG. 3.11: En gris W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 2$ et $\theta = -0.1$ rad

pouvons observer que plus τ_{max} est petit, plus la surface de W_s diminue mais nous remarquerons surtout la disparition et l'apparition de différentes régions constitutives de W_s . Ainsi comme sur la figure 3.9, W_s est toujours constitué de trois régions sur la figure 3.10 mais la région 3 de 3.9 diminue de taille (figure 3.10) puis disparaît (figure 3.11). Si nous continuons à faire décroître la valeur de τ_{max} , nous observons que les régions 1 et 2 se séparent en deux composantes et l'on compte alors quatre régions distinctes pour W_s (figures 3.12 et 3.13). Remarquons également que sur la figure 3.13 les quatre composantes ne sont plus reliées entre elles par des points singuliers puisque l'on observe qu'une région de W_s s'est séparée des trois autres.

Un autre exemple intéressant est illustré sur la figure 3.14. Nous pouvons y voir l'espace de travail statique composé de quatre régions qui sont reliées l'une à l'autre par trois

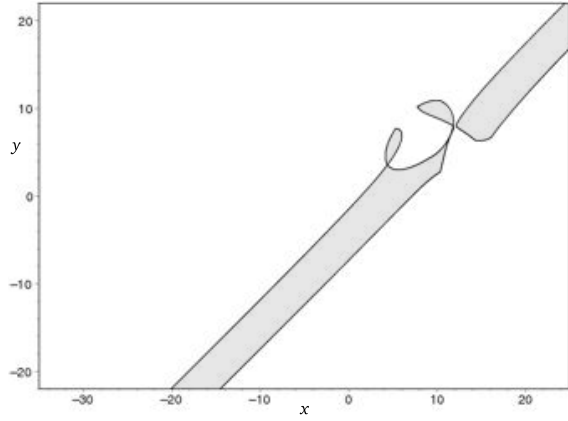


FIG. 3.12: En gris W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 1.5$ et $\theta = -0.1 \text{ rad}$

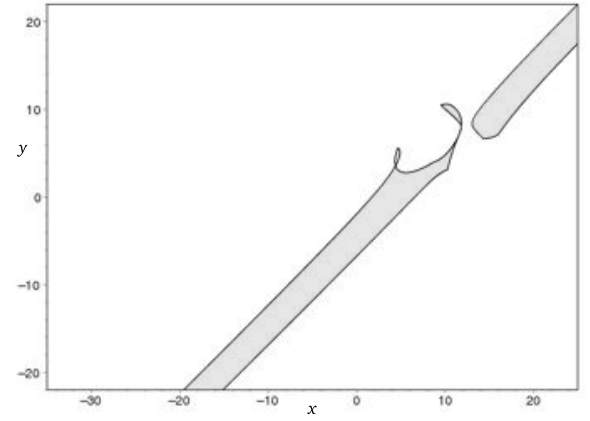


FIG. 3.13: En gris W_s pour $\mathcal{F} = (1, 1, 1)$, $\tau_{max} = 1.3$ et $\theta = -0.1 \text{ rad}$

points singuliers. Pour passer d'une composante de W_s à une autre il n'existe même qu'un seul point (voir figure 3.14). Ces trois points singuliers, reliant les diverses composantes entre elles, correspondent aux cas où l'une des longueurs de jambe du robot est nulle ($\rho_i = 0$). Si nous imposons en plus les contraintes $\forall i = 1, n \rho_i \geq 4$, les quatre composantes de l'espace de travail statique deviennent disjointes (figure 3.15). Sur cette figure nous avons également tracé en pointillé les lieux des singularités qui, dans ce cas, sont les droites d'équations $y = 4$ et $y = -2.5x + 38$.

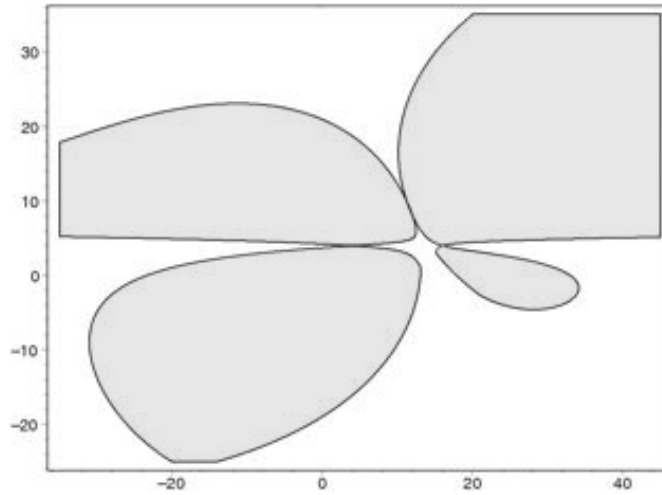


FIG. 3.14: En gris W_s pour $\mathcal{F} = (4, 0, 0)$, $\tau_{max} = 1.3$ et $\theta = -0.1 \text{ rad}$

3.6.2 Temps de calcul

Le temps de calcul pour la détermination de toutes les courbes varie en règle général entre 800 et 1500 secondes avec un processeur *Intel(R) Core(TM)2 CPU 2.40GHz*. Mais comme il n'est pas nécessaire de les calculer entièrement, nous obtenons généralement l'espace de travail statique pour des temps de calcul compris entre 100 et 300 secondes. Ces variations s'expliquent simplement par le nombre d'arcs de courbes nécessaires pour

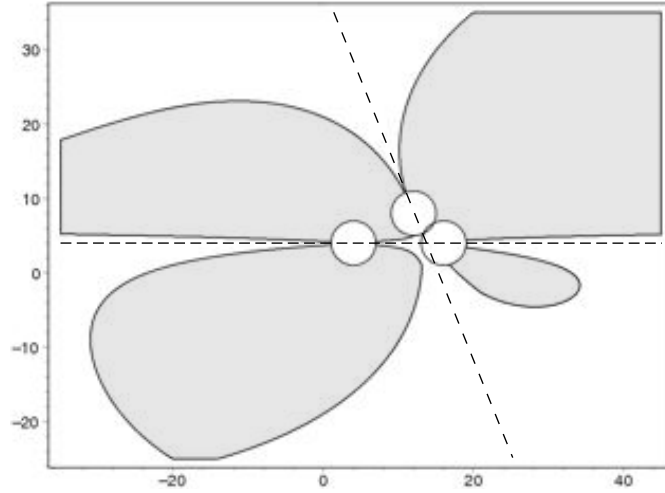


FIG. 3.15: En gris W_s pour $\mathcal{F} = (4, 0, 0)$, $\tau_{max} = 1.3$ et $\theta = -0.1 \text{ rad}$, avec l'imposition d'une contrainte géométrique sur la longueur des jambes ($\rho_{min} = 4$) et en pointillé les singularités qui sont définies par les droites d'équations $y = 4$ et $y = -2.5x + 38$.

obtenir l'espace de travail statique. L'utilisation du C++ conduirait certainement à une diminution notable de ces temps de calcul.

3.7 Conclusion

Dans ce chapitre, nous avons d'abord défini un index de proximité aux singularités qui repose sur le critère physique suivant : pour éviter de menacer l'intégrité physique d'un robot, il faut que les efforts dans les jambes de ce dernier soient inférieurs à un seuil donné. Nous avons choisi ce seuil comme la force ou le couple de rupture du plus faible composant mécanique de la chaîne.

A partir de cet index, nous avons défini l'espace de travail statique W_s comme une zone sûre au sens statique, c'est à dire l'ensemble des poses du robot pour lesquelles l'intégrité physique du mécanisme n'est pas menacée.

Enfin nous nous sommes attachés à calculer l'espace de travail statique. Nous avons décrit un algorithme permettant de calculer des sections planes de l'espace de travail statique pour un mécanisme donné, l'orientation de la plate-forme et la charge étant fixées. Nous avons donc obtenu une approximation polygonale de la frontière Γ_s de W_s , ceci avec une précision arbitraire. Cette approximation polygonale a l'avantage de permettre le calcul de la surface de l'espace de travail statique, par exemple à des fins de comparaison entre robots.

Enfin, quoique notre algorithme ait été illustré sur un robot particulier (robot plan 3 – RPR), il est possible de l'adapter pour obtenir des approximations polygonales de Γ_s pour d'autres types de manipulateurs plans.

Cet algorithme a cependant le défaut de ne produire que des sections planes de W_s pour une charge donnée. Le but du chapitre suivant est d'introduire les prérequis nécessaires à la compréhension d'un autre algorithme, qui permettra d'obtenir une approximation de l'espace de travail statique complet, ceci pour un ensemble de charges.

INTRODUCTION À L'ANALYSE PAR INTERVALLES

4.1 Introduction

Dans le chapitre précédent nous avons pu établir des coupes de l'espace de travail statique pour une charge donnée. Toutefois il pourrait être intéressant de calculer l'espace de travail statique dans sa globalité et pour un ensemble de charges. C'est ce que nous ferons dans le chapitre 5 en utilisant une approche reposant sur *l'analyse par intervalles* que nous introduisons dans ce chapitre.

La représentation des nombres pose généralement un problème car il faut pouvoir décrire ce nombre de manière finie. Si nous nous attachons à une représentation décimale, le plus célèbre exemple est sûrement le nombre π qui ne peut être représenté de manière exacte. Ainsi ce nombre est souvent approximé par $\pi \approx 3.14$ ce qui conduit à une erreur d'arrondi. Nous devrions donc dire que π vaut 3.14 à un nombre ϵ près. Une manière plus précise d'approximer π avec l'écriture décimale consiste à le définir grâce au plus petit des majorants représentables et au plus grand des minorants représentables. Ainsi, si l'on cherche une approximation avec cinq chiffres significatifs on pourra dire que $\pi \in [3.1415, 3.1416]$. Nous venons donc d'encadrer le nombre π dans l'intervalle réel fermé borné $[3.1415, 3.1416]$.

L'analyse par intervalles naquit ainsi dans les années 1960 du besoin qu'avaient les numériciens de certifier leurs calculs, ou plus généralement de garantir une précision. Cette affirmation peut surprendre car l'analyse numérique a déjà apportée de nombreux résultats concernant la résolution numérique de systèmes d'équations. En règle générale, ces résultats sont le fruit de multiples opérations, chacune garantie à une précision donnée, mais sans garantie de précision sur le résultat de l'ensemble de ces opérations.. Le meilleur moyen de s'en convaincre est de s'intéresser à l'exemple très célèbre de Rump [48].

Soit la fonction :

$$f(x, y) = (333.75 - x^2)y^6 + (x^2(11x^2y^2 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y}) \quad (4.1)$$

qui pour les valeurs particulières $x_0 = 77617$ et $y_0 = 33096$ se simplifie sous la forme :

$$f_0(x_0, y_0) = \frac{x_0}{2y_0} - 2 \quad (4.2)$$

Valeur de " <i>Digits</i> "	Résultats donnés par Maple pour $f(x_0, y_0)$
15	$-0.2 \cdot 10^{23}$
30	$0.200000011726039400531786318588 \cdot 10^8$
33	$-9998.82739605994682136814116509548 \cdot 10^{33}$
34	$-1998.827396059946821368141165095480$
35	$-198.82739605994682136814116509547982$
36	$21.1726039400531786318588349045201837$
37	$-0.827396059946821368141165095479816292$

TAB. 4.1: Evaluation de la fonction 4.1 pour $x = 77617$ et $y = 33096$ selon le nombre de "*Digits*"

Le tableau (4.1) présente le résultat de l'évaluation de (4.1) pour les valeurs particulières $x = x_0$ et $y = y_0$ sous *Maple*, selon le nombre de chiffres significatifs souhaités, que l'on obtient en modifiant la variable "*Digits*".

En utilisant (4.2) nous obtenons $f_0(x_0, y_0) = -0.82739\dots$, alors que la table montre des variations très importantes de l'évaluation avec (4.1) et ce n'est qu'en augmentant fortement la précision (*Digits*=37) que nous obtenons une évaluation qui est *proche* du véritable résultat.

C'est pour faire face à ce problème qu'a été initialement introduit l'analyse par intervalle, que nous allons introduire dans les sections suivantes. Nous verrons cependant ultérieurement que l'analyse par intervalles permet bien plus que de travailler sur la précision des calculs numériques.

4.2 Définitions élémentaires

Définition : Un intervalle X est défini par ses bornes inférieure \underline{x} et supérieure \bar{x} . On désignera désormais par *intervalle*, l'intervalle réel fermé borné défini par :

$$X = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\} = [\underline{x}, \bar{x}]$$

Il existe d'autres manières de définir des intervalles mais nous ne les aborderons pas. On désignera par \mathbb{IR} l'ensemble des intervalles réels.

Désignons par *intervalle dégénéré*, l'intervalle dont la borne supérieure et la borne inférieure sont égales. Nous noterons désormais l'intervalle dégénéré $X = [x, x]$ par x .

Nous allons maintenant nous attacher à définir quelques fonctions qui permettent de caractériser un intervalle $X \in \mathbb{IR}$:

- la fonction appliquée à un intervalle X donnant comme résultat sa borne inférieure sera notée $\inf(X) = \underline{x}$ et celle qui donne la borne supérieure $\sup(X) = \bar{x}$

• le *milieu* ou *centre* d'un intervalle se définit par : $mid(X) = \frac{\underline{x} + \bar{x}}{2}$. Nous trouverons également dans la littérature la notation \check{x}

• la *largeur* d'un intervalle X est notée $w(X)$ (w pour *width*) et se définit par $w(X) = \bar{x} - \underline{x}$.

Remarquons que par la définition même d'un intervalle, la largeur est toujours positive (nulle dans le cadre d'un intervalle dégénéré).

• le *rayon* $rad(X)$ (rad pour *radius*) d'un intervalle X est la moitié de la largeur d'un intervalle : $rad(X) = \frac{\bar{x} - \underline{x}}{2}$.

Remarquons que pour n'importe quel point $x \in X$ nous pouvons dire que $|x - mid(X)| \leq rad(X)$ et réciproquement. Ainsi le rayon d'un intervalle X est un critère qui mesure l'erreur maximum entre le point milieu considéré ici comme une approximation d'un point quelconque x de l'intervalle.

On définit aussi :

• la *magnitude* d'un intervalle X est notée $mag(X)$ et se définit par $mag(X) = max(|\underline{x}|, |\bar{x}|)$

• la *mignitude* d'un intervalle X se définit comme la valeur minimale de $|x|$ pour tout $x \in X$ donc :

$$mig(x) = \begin{cases} \underline{x} & \text{si } \underline{x} > 0 \\ \bar{x} & \text{si } \bar{x} < 0 \\ 0 & \text{sinon} \end{cases}$$

ce qui permet de définir la valeur absolue d'un intervalle $|X|$ par $|X| = [mig(X), mag(X)]$

Définition : Par analogie avec l'espace \mathbb{R}^n qui désigne l'espace des vecteurs réels $\mathbf{x} = (x_1 \dots x_n)$, définissons l'espace \mathbb{IR}^n comme l'espace des vecteurs intervalles $\mathbf{x}^I = (X_1 \dots X_n)$. Désignons par *vecteur intervalle* ou plus simplement *boîte* (*Box*) l'ensemble des vecteurs définis par :

$$\mathbf{x}^I = \{\mathbf{x} \in \mathbb{R}^n : \forall i = 1..n \quad x_i \in X_i\}$$

De manière similaire on définira les fonctions *inf*, *sup*, *mid*, *rad*, *mig*, *mag* et $|| \cdot ||$ composantes par composantes ainsi :

- $\underline{\mathbf{x}} = inf(\mathbf{x}^I) = (\underline{x}_1 \dots \underline{x}_n)$
- $\bar{\mathbf{x}} = sup(\mathbf{x}^I) = (\bar{x}_1 \dots \bar{x}_n)$
- $rad(\mathbf{x}^I) = (rad(X_1) \dots rad(X_n))$
- $|\mathbf{x}^I| = (|X_1| \dots |X_n|)$
- $mig(\mathbf{x}^I) = (mig(X_1) \dots mig(X_n))$
- $mag(\mathbf{x}^I) = (mag(X_1) \dots mag(X_n))$

On ajoutera également les définitions de la *norme d'un vecteur intervalle* et du *diamètre* de la boîte :

- $||\mathbf{x}^I|| = max(|X_i|, i = 1..n)$

$$\cdot \text{diam}(\mathbf{x}^{\mathbf{I}}) = \max(w(X_i), i = 1..n)$$

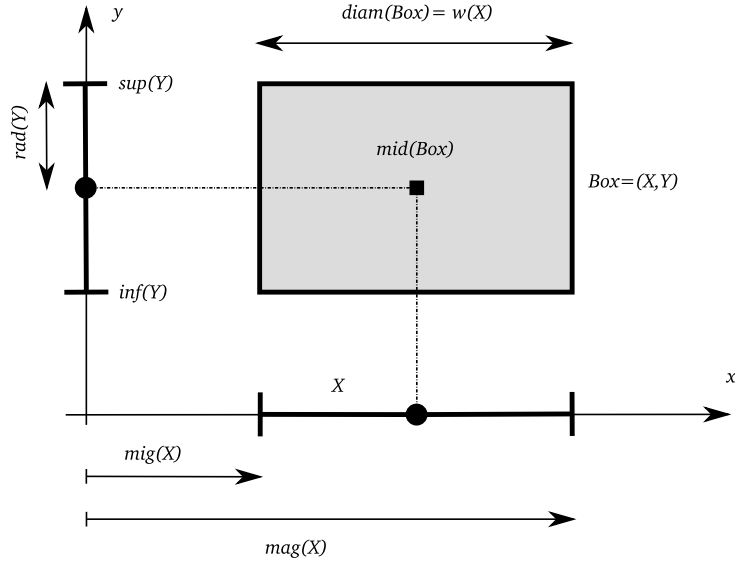


FIG. 4.1: Récapitulatif des définitions

Définition : Une *matrice intervalle* $\mathbf{A}^{\mathbf{I}}$ de dimension $m \times n$ est un tableau d'intervalles de m lignes et n colonnes. Nous noterons A_{ij} l'intervalle d'un telle matrice correspondant à la i -ème ligne et la j -ème colonne du tableau. Plus formellement on considérera une matrice intervalle comme l'ensemble des matrices définies par

$$\mathbf{A}^{\mathbf{I}} = \{\mathbf{A} \in \mathbb{IR}^{n \times m} : a_{ij} \in A_{ij}\}$$

Comme pour les vecteurs, nous définis les fonctions *inf*, *sup*, *mid*, *rad*, *mig*, *mag* et $||$ composantes par composantes.

Définition : Nous dirons qu'une n matrice carrée intervalle $\mathbf{A}^{\mathbf{I}}$ est régulière si toutes les matrices réelles \mathbf{A} qu'elle contient sont régulières. A l'opposé, une matrice intervalle sera dite singulière si parmi toutes les matrices réelles contenues dans l'ensemble $\mathbf{A}^{\mathbf{I}}$, il en existe au moins une singulière.

4.2.1 Théorie des ensembles et intervalles

Comme pour n'importe quel ensemble, les opérations d'union et d'intersection sont définies pour les intervalles.

L'*intersection* \cap de deux intervalles X et Y est définie par :

$$X \cap Y = \{z \in X \text{ et } z \in Y : \max(\underline{x}, \underline{y}) \leq z \leq \min(\bar{x}, \bar{y})\}$$

Si les bornes des deux intervalles X et Y sont telles que $\bar{x} < \underline{y}$ ou $\bar{y} < \underline{x}$, alors l'intersection sera vide : $X \cap Y = \emptyset$

Remarque : Attirons l'attention du lecteur sur l'importance de l'intersection dans la résolution de problème numérique. En effet si un résultat appartient à deux intervalles différents alors le résultat appartient forcément à l'intersection de ces deux intervalles.

L'*union* \cup de deux intervalles dont l'intersection est non vide sera l'intervalle

$$X \cup Y = \{z \in X \text{ ou } z \in Y : \min(\underline{x}, \underline{y}) \leq z \leq \max(\bar{x}, \bar{y})\}$$

Cependant si l'intersection est vide, l'union de deux intervalles ne forme pas forcément un intervalle. On peut toutefois définir le plus petit intervalle qui va contenir cette union. Il se nomme l'*enveloppe intervalle* et il se définit comme :

$$Hull(X, Y) = \{z \in \mathbb{R} : \min(\underline{x}, \underline{y}) \leq z \leq \max(\bar{x}, \bar{y})\}$$

qui pour deux intervalles X et Y quelconque vérifiera toujours $X \cup Y \subseteq Hull(X, Y)$.

4.2.2 Arithmétique par intervalles

Le but de l'arithmétique par intervalles est de définir des opérations élémentaires qui appliquées à des intervalles X_1 et X_2 vont retourner un intervalle X . Cet intervalle contiendra forcément pour la même opérations les résultats fournis par n'importe quel couple de scalaires (x_1, x_2) contenus dans (X_1, X_2) .

4.2.2.1 Opérations usuelles

:

Les opérations arithmétiques usuelles peuvent être étendues aux intervalles mais on y trouvera cependant quelques différences par rapport aux opérations sur les réels. Ainsi les opérations classiques comme l'addition, la soustraction, la multiplication et la division seront définies pour deux intervalles donnés $X = [\underline{x}, \bar{x}]$ et $Y = [\underline{y}, \bar{y}]$ par :

- $X + Y = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$
- $X - Y = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$
- $X * Y = [\min(\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y}), \max(\underline{x} * \underline{y}, \underline{x} * \bar{y}, \bar{x} * \underline{y}, \bar{x} * \bar{y})]$
- $X / Y = [\min(\underline{x} / \underline{y}, \underline{x} / \bar{y}, \bar{x} / \underline{y}, \bar{x} / \bar{y}), \max(\underline{x} / \underline{y}, \underline{x} / \bar{y}, \bar{x} / \underline{y}, \bar{x} / \bar{y})]$ si $0 \notin Y$

Remarque : On s'aperçoit que du point de vue intervalle $X - X \neq 0$. Cela illustre l'importance de trouver la formulation la plus appropriée quand on travaille dans le cadre de l'arithmétique par intervalles. Plus généralement on remarquera que la soustraction n'est pas l'opération réciproque de l'addition et que la division n'est pas l'opération réciproque de la multiplication comme c'est le cas par exemple pour les réels. En effet $[3, 5] - [1, 2] = [1, 6]$ et $[1, 6] + [1, 2] = [2, 8] \neq [3, 5]$.

Concernant la division quand $0 \in Y$, plusieurs considérations s'affrontent au niveau numérique. La première consiste à considérer l'opération comme impossible car pour un

des éléments de l'intervalle Y à savoir 0, la division n'est pas définie et nous considérerons que l'opération n'a pas lieu d'être. Ainsi le résultat sera considéré comme un NAI ("not an interval"). Pour d'autres l'intervalle $[-\infty, +\infty]$ sera considéré comme résultat.

Enfin la division étendue a également été définie et a la particularité de renvoyer deux résultats mais ceci nous fait sortir du cadre de l'arithmétique par intervalles :

- $[-\infty, \overline{res}]$ avec $\overline{res} \leq 0$
- $[\underline{res}, +\infty]$ avec $\underline{res} \geq 0$

Cependant, dans le cadre de la division étendue, le résultat de l'opération $[1, 1]/[0, 1]$ sera limité au seul intervalle $[1, +\infty]$

• Muni de ces opérations usuelles, nous pourrions définir par analogie avec les réels des opérations telles que le produit scalaire de deux vecteurs intervalles, le produit matrice vecteur ou le produit matriciel. Remarquons à propos du produit matrice vecteur intervalle $\mathbf{A}^{\mathbf{I}}\mathbf{x}^{\mathbf{I}} = \sum A_{ij}X_j$ qu'il est différent de l'ensemble $\{A\mathbf{x} : \mathbf{A} \in \mathbf{A}^{\mathbf{I}}, \mathbf{x} \in \mathbf{x}^{\mathbf{I}}\}$ si la matrice $\mathbf{A}^{\mathbf{I}}$ est une matrice intervalle dégénérée.

Lois de composition interne :

Avant toute chose, rappelons la définition de l'inclusion entre deux intervalles :

- $X \subseteq Y \iff \underline{x} \geq \underline{y} \text{ et } \overline{x} \leq \overline{y}$
- $X \subset Y \iff \{\underline{x} > \underline{y} \text{ et } \overline{x} \leq \overline{y}\} \text{ ou } \{\underline{x} \geq \underline{y} \text{ et } \overline{x} < \overline{y}\}$

Nous étendons ces définitions pour les boîtes intervalles, en raisonnant composantes par composantes.

Si on travaille sur \mathbb{IR} , alors comme pour les réels les lois d'associativité et de commutativité pour l'addition et la multiplication sont vérifiées :

$\forall X, Y, Z \in \mathbb{IR}$

- $X + Y = Y + X$
- $X + (Y + Z) = (X + Y) + Z$
- $XY = YX$
- $X(YZ) = (XY)Z$

Cependant comparativement aux réels, nous perdons la propriété de distributivité mais nous pouvons tout de même définir la sous-distributivité :

$$\bullet X(Y + Z) \subseteq XY + XZ$$

Nous obtenons la distributivité dans le cas où X est un intervalle dégénéré. Remarquons une nouvelle fois l'importance d'une formulation appropriée quand nous cherchons à obtenir une évaluation la plus précise possible.

Précisons que pour les matrices intervalles par exemple on ne vérifiera pas les propriétés suivantes contrairement au cas réel :

- $\forall \mathbf{A}^{\mathbf{I}}, \mathbf{B}^{\mathbf{I}}, \mathbf{C}^{\mathbf{I}} \in \mathbb{IR}^{n \times n} \quad \mathbf{A}^{\mathbf{I}}(\mathbf{B}^{\mathbf{I}}\mathbf{C}^{\mathbf{I}}) \neq (\mathbf{A}^{\mathbf{I}}\mathbf{B}^{\mathbf{I}})\mathbf{C}^{\mathbf{I}}$
- $\forall \mathbf{A}^{\mathbf{I}} \in \mathbb{IR}^{n \times n}, \mathbf{x}^{\mathbf{I}} \in \mathbb{IR}^n, \alpha \in \mathbb{R} \quad \mathbf{A}^{\mathbf{I}}(\alpha\mathbf{x}^{\mathbf{I}}) \neq \alpha(\mathbf{A}^{\mathbf{I}}\mathbf{x}^{\mathbf{I}})$

4.2.3 Fonctions et intervalles

Définition : Soit une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. On désigne par l'image $f(\mathbf{x}^I)$ de la fonction f sur la boîte \mathbf{x}^I l'ensemble des images de la fonction f sur la boîte \mathbf{x}^I

$$f(\mathbf{x}^I) = \{f(\mathbf{x}) : \mathbf{x} \in \mathbf{x}^I\}$$

Remarquons que l'image de la fonction sur la boîte n'est généralement pas un intervalle (sauf si $m = 1$).

Définition : désignons donc par *fonction d'évaluation* de la fonction f , la fonction $f^I : \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ telle que :

$$\begin{aligned} \bullet \forall \mathbf{x} \in \mathbb{R}^n \quad f^I(\mathbf{x}) &= f(\mathbf{x}) \\ \bullet \forall \mathbf{x}^I \in \mathbb{IR}^n \quad f^I(\mathbf{x}^I) &\supseteq f(\mathbf{x}^I) \end{aligned}$$

Nous trouvera également dans la littérature, les termes *fonction intervalle* ou *extension intervalle* pour désigner la fonction d'évaluation.

Si $m = 1$ nous avons donc pour une boîte $\mathbf{x}^I = (X_1 \dots X_n)$ l'image de la fonction intervalle qui vaut $f^I(X_1 \dots X_n) = [\underline{f}^I, \overline{f}^I]$. On pourra alors dire :

$$\bullet \forall x_i \in X_i \quad i = 1..n \quad \underline{f}^I \leq f(x_1 \dots x_n) \leq \overline{f}^I.$$

Autrement dit la fonction d'évaluation renvoie un minorant et un majorant de l'image de la fonction f sur la boîte \mathbf{x}^I .

Certains définissent aussi l'extension optimale f^\square d'une fonction f et l'image $f^\square(x^I)$ de cette extension optimale sur la boîte \mathbf{x}^I comme la plus petite boîte contenant l'image de la fonction $f(\mathbf{x}^I)$. Précisons toutefois que le calcul de cette extension optimale est un problème NP-dur (non déterministe polynôme-temps dur) dès que l'expression de f devient compliquée [30]. C'est donc un objectif théorique qui pourra toutefois être atteint dans certains cas. Un des enjeux majeurs de l'analyse par intervalles est de trouver la meilleure extension intervalle d'une fonction.

Définition : Une fonction intervalle f^I est dite *monotone pour l'inclusion* si elle vérifie :

$$\bullet \mathbf{x}^I \subset \mathbf{y}^I \Rightarrow f^I(\mathbf{x}^I) \subset f^I(\mathbf{y}^I)$$

Théorème Fondamental : N'importe quelle fonction dont l'évaluation utilise une séquence d'opérations n'impliquant que l'addition, la soustraction, la multiplication et la division est monotone inclusive.

Un intérêt pratique de ce théorème est la prise en compte des erreurs numériques. Ainsi toute valeur non représentable par un ordinateur sera contenue dans l'évaluation intervalle. Pratiquement des nombres comme $\pi, \frac{1}{3}$ ou même 0.1, qui n'ont pas de représentation exacte sur un ordinateur, seront représentés par un intervalle dont les bornes seront le plus grand flottant inférieur à $\pi, \frac{1}{3}$ ou 0.1 et le plus petit flottant supérieur à leurs valeurs respectives. Ainsi la multiplication de $\frac{1}{3}$ par 3 sera un intervalle qui

contient 1.

Un autre conséquence de ce théorème concerne la résolution d'équations, si l'évaluation intervalle $f^I(X)$ d'une fonction f est strictement positive ou négative alors on peut certifier qu'il n'existe pas de solution à l'équation $f(x) = 0$ où $x \in X$.

4.3 Un enjeu primordial : l'évaluation

L'évaluation d'une fonction est sûrement l'un des enjeux les plus importants en analyse par intervalles car elle est présente à chaque étape et son efficacité a donc une influence significative sur le temps de calcul. On va ici rappeler quels peuvent être les différents moyens d'améliorer l'évaluation $f^I(X)$ d'une fonction f sur un intervalle X .

En préambule, nous allons expliquer ce que l'on entend par *fonction d'évaluation naturelle*. Elle repose sur le fait que chaque fonction mathématique élémentaire (*cos*, *sin*, *tan*, *exp*, *ln*, etc...) possède un équivalent intervalle. Aussi en pratique, pour évaluer naturellement une fonction $f(x)$ sur un domaine donnée X , nous remplaçons chaque variable par son intervalle. Cette méthode donne un résultat qui peut surestimer le résultat réel. Nous allons donc nous attacher dans un premier temps à présenter les causes de surestimation de l'évaluation puis différentes méthodes permettant d'améliorer l'évaluation.

4.3.1 Les causes d'évaluation non optimale

Nous avons d'ores et déjà abordé certains problèmes inhérents à l'analyse par intervalles dans des remarques précédentes. Cependant par souci pédagogique et afin de bien comprendre quelles sont les difficultés liées à l'évaluation des fonctions, nous allons nous intéresser aux deux difficultés majeures que sont *l'effet enveloppant* et les *occurrences multiples de variables*.

4.3.1.1 L'effet enveloppant

Nous avons déjà fait remarquer que l'image d'une fonction $f(\mathbf{x}^I)$ était différente de son extension intervalle $f^I(\mathbf{x}^I)$. Comme l'extension intervalle d'une fonction revient à calculer un majorant et un minorant de l'image d'une fonction on pourra même dire que $f(\mathbf{x}^I) \subseteq f^I(\mathbf{x}^I)$. C'est ce qu'on dénote par *effet enveloppant*. Un bon moyen d'illustrer l'effet enveloppant est de s'intéresser par exemple à une fonction élémentaire :

$$\begin{aligned} f : \mathbb{R} &\rightarrow \mathbb{R}^2 \\ x &\mapsto (x, 2x) \end{aligned}$$

Si nous regardons l'image $f(X)$ de la fonction f sur l'intervalle $X = [1, 2]$, nous obtenons une portion de la droite d'équation $y = 2x$. Cependant si on s'intéresse à l'extension intervalle $f^I(X)$ de la fonction sur ce même intervalle, nous allons obtenir comme solution la boîte $([1, 2], [1, 4])$ (en gris sur la figure 4.2). Nous allons donc largement surestimer les valeurs possibles de la fonction sur l'intervalle considéré. Un des moyens de réduire ce volume sera de subdiviser l'intervalle initiale X en une multitude de sous-intervalles X_i tels que $X = \bigcup X_i$ (voir la figure 4.2). Il n'en résulte pas moins que pour chacun de ces différents sous-intervalles, nous aurons toujours un effet enveloppant.

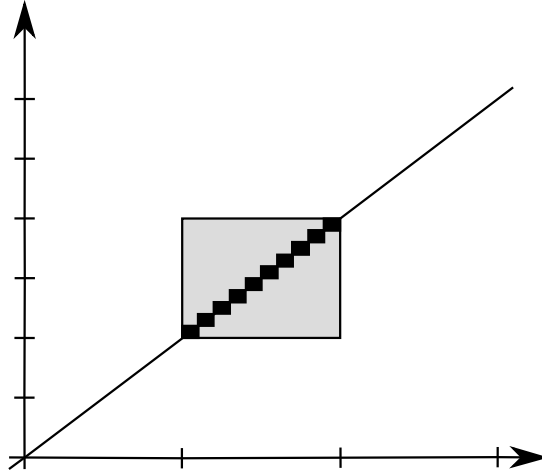


FIG. 4.2: Evaluation de la fonction $y = 2x$ pour l'intervalle $[1, 2]$ en grisé et pour des sous-intervalles dont l'union est $[1, 2]$ en noir

4.3.1.2 Occurrences multiples de variables

L'autre problème majeur de l'analyse par intervalle est la perte de la dépendance des données qui peut conduire à une nette surévaluation d'une fonction. Un bon moyen d'illustrer ce phénomène est de s'intéresser à la fonction réelle élémentaire $f : x \mapsto x$ et à différentes manières d'écrire cette dernière :

- $f_1(x) = x$
- $f_2(x) = 2x - x$

Si nous nous intéressons à l'évaluation de ces fonctions sur l'intervalle $X = [1, 2]$, nous obtiendrons comme résultat :

- $f_1^I(X) = X = [1, 2]$
- $f_2^I(X) = 2X - X = [2, 4] - [1, 2] = [0, 3]$

Nous nous étonnerons (seulement légèrement car on a déjà abordé le sujet avec la notion de sous-distributivité) que l'évaluation de f_2 soit plus large que celle de f_1 . En fait c'est simplement dû au fait que lors de l'évaluation de f_2^I le calcul suivant est effectué :

- $f_2^I(X, X') = 2X - X'$ où $X = [1, 2]$ et $X' = [1, 2]$

nous venons de perdre la relation $X = X'$ et certains parlent parfois de *perte de la dépendance des données*.

Remarque : Nous pouvons également voir l'effet enveloppant comme un corollaire de la perte de la dépendance des données. En effet si nous revenons à l'exemple précédent, nous pouvons voir que pour la fonction précédente, nous effectuons le calcul suivant : $f^I : (X, X') \mapsto (X, 2X')$ avec $X = [1, 2]$ et $X' = [1, 2]$ et nous avons, à nouveau, perdu la relation $X = X'$. C'est pour cela que nous entendons parfois parler de *perte de dépendances des données* pour expliquer ces deux phénomènes [41].

4.3.2 Utilisation de la formule de Taylor dans le cas de l'analyse par intervalles

Rappelons la formule de Taylor-Lagrange sous réserve que la fonction f soit bien différentiable.

$$f(y) = f(x) + (y-x)f'(x) + \dots + \frac{(y-x)^n}{n!}f^n(x) + \frac{(y-x)^{n+1}}{(n+1)!}f^{n+1}(\xi) \quad \text{avec } \xi \in [x, y] \quad (4.3)$$

Si x et y sont inclus dans un intervalle X alors nous pourrions borner le terme $f^{n+1}(\xi)$ par son évaluation par intervalle $(f^I)^{n+1}(X)$, ce qui permet d'écrire

$$f(y) \in f(x) + (y-x)f'(x) + \dots + \frac{(y-x)^n}{n!}f^n(x) + \frac{(y-x)^{n+1}}{(n+1)!}(f^I)^{n+1}(X) \quad (4.4)$$

Ecrivons par exemple

- pour le cas $n = 0$

$$\forall x \in X \quad f^I(X) \subset f(x) + (X-x)(f^I)'(X) \quad (4.5)$$

- pour le cas $n = 1$

$$\forall x \in X \quad f^I(X) \subset f(x) + (X-x)(f^I)'(x) + \frac{(X-x)^2}{2}(f^I)''(X) \quad (4.6)$$

Insistons sur le fait que cette expression est à prendre strictement au sens intervalle, car bien que ξ et y soient tous deux compris dans un même intervalle X , les termes $f^I(X)$ et $(f^I)'(X)$ (pour le cas $n=0$) ne peuvent être associés puis simplifiés. En effet il s'agit de deux intervalles différents X_y et X_ξ qui sont tous deux égales à X .

Si au lieu d'un point quelconque $x \in X$, on choisit le point milieu $mid(X)$ on parle alors de *forme centrée*.

Cas Multidimensionnel : Soient \mathbf{x} et \mathbf{y} deux vecteurs de dimension m tels que leurs composantes x_i et y_i soient incluses dans les composantes X_i d'une boîte \mathbf{x}^I de dimension m .

De manière analogue au cas unidimensionnel, nous sommes tenté d'écrire pour le cas $n = 0$

$$f^I(\mathcal{X}) \subset f(\mathbf{x}) + (\mathbf{x}^I - \mathbf{x})\nabla f^I(\mathbf{x}^I) \quad (4.7)$$

Bien que cette formulation soit exacte, il a été montré qu'elle peut être cependant améliorée [20]. Montrons comment pour un cas ou $m = 3$ puis on en déduira la cas m -dimensionnel.

Nous utilisons Taylor variable par variable en commençant par la troisième, puis la seconde et enfin la première ce qui donne :

$$f(y_1, y_2, y_3) = f(y_1, y_2, x_3) + (y_3 - x_3)\partial_3 f(y_1, y_2, \xi_3) \quad (4.8)$$

$$f(y_1, y_2, x_3) = f(y_1, x_2, x_3) + (y_2 - x_2)\partial_2 f(y_1, \xi_2, x_3) \quad (4.9)$$

$$f(y_1, x_2, x_3) = f(x_1, x_2, x_3) + (y_1 - x_1)\partial_1 f(\xi_1, x_2, x_3) \quad (4.10)$$

En combinant ces trois équations, nous obtenons

$$\begin{aligned} f(\mathbf{y}) = f(\mathbf{x}) &+ (y_1 - x_1)\partial_1 f(\xi_1, x_2, x_3) \\ &+ (y_2 - x_2)\partial_2 f(y_1, \xi_2, x_3) \\ &+ (y_3 - x_3)\partial_3 f(y_1, y_2, \xi_3) \end{aligned} \quad (4.11)$$

Par analogie avec le travail unidimensionnel nous déduisons quelque soient x et y contenus dans \mathbf{x}^I

$$\begin{aligned} f(\mathbf{y}) \in f(\mathbf{x}) &+ (y_1 - x_1)\partial_1 f^I(X_1, x_2, x_3) \\ &+ (y_2 - x_2)\partial_2 f^I(y_1, X_2, x_3) \\ &+ (y_3 - x_3)\partial_3 f^I(y_1, y_2, X_3) \end{aligned} \quad (4.12)$$

Donc nous pouvons écrire pour le cas m -dimensionnel :

$$f^I(\mathbf{X}) \subset f(\mathbf{x}) + \sum_{i=1}^m (X_i - x_i)\partial_i f^I(X_1, \dots, X_i, x_{i+1}, \dots, x_m) \quad (4.13)$$

4.3.3 Utilisation de la monotonie

Si une fonction f est monotone sur un intervalle $X = [x_{min}, x_{max}]$, alors la meilleure évaluation possible de f sur X sera donnée par

$$f^I(X) = [\min(f(x_{min}), f(x_{max})), \max(f(x_{min}), f(x_{max}))].$$

Cas Multidimensionnel : Soit $\mathbf{x}^I = (X_1, \dots, X_n)$ une boîte de dimension n , nous cherchons à évaluer la fonction $f(x_1, \dots, x_n)$ sur \mathbf{x}^I . Nous allons donc dans un premier temps calculer les dérivées $\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}$. Nous allons ensuite les évaluer sur \mathbf{x}^I et regarder quelles sont celles qui ont éventuellement un signe constant. Recherchons les variables x_i pour lesquelles la fonction $f(x_i)$ est strictement monotone, c'est à dire $\frac{\partial f^I}{\partial x_i}(X_1, \dots, X_n) > 0$ ou < 0 .

Par exemple si nous supposons que $\frac{\partial f^I}{\partial x_i}(X_1, \dots, X_n) > 0$ pour $i = 1..p$ et

$$\frac{\partial f^I}{\partial x_i}(X_1, \dots, X_n) < 0 \text{ pour } i = p+1..m.$$

Nous pouvons dès lors affirmer : le maximum de f est inclus dans

$$f^I(x_{1_{max}}, \dots, x_{p_{max}}, x_{p+1_{min}}, \dots, x_{m_{min}}, X_{m+1}, \dots, X_n)$$

Il va de soit qu'avec un procédé analogue, nous pourrions également trouver un bon encadrement du minimum. Cependant, nous pouvons encore améliorer l'évaluation. Comme nous cherchons à minimiser le nombre d'intervalles pris en compte pour l'évaluation de la fonction f^I , nous allons évaluer de manière récursive les dérivées partielles en tenant

compte des résultats précédents.

En effet la dérivée partielle $\frac{\partial f}{\partial x_i}$ (avec $i = m + 1..n$) qui originellement était de signe non constant peut très bien, en tenant compte des résultats précédents, devenir positive, négative voire nulle.

Comme nous l'avons déjà fait remarquer, nous aurons généralement une meilleure évaluation avec $\frac{\partial f_{max}^I}{\partial x_i}(X_m + 1, \dots, X_n)$ qu'avec $\frac{\partial f^I}{\partial x_i}(X_1, \dots, X_n)$.

On a $f_{max}^I(X_m + 1, \dots, X_n) = f^I(x_{1_{max}}, \dots, x_{p_{max}}, x_{p+1_{min}}, \dots, x_{m_{min}}, X_{m+1}, \dots, X_n)$ dans notre cas.

Nous évaluera donc les dérivées partielles de manière récursive, nous nous arrêterons pour le calcul des bornes supérieures quand on aura soit tous les intervalles remplacés par des scalaires, soit quand le nombre de dérivées partielles de signe constant reste le même et ce malgré la prise en compte de scalaires en lieu et place des intervalles.

Remarquons que lors de l'évaluation d'une fonction on pourra tout à fait utiliser ce processus reposant sur la monotonie en l'appliquant par exemple aux dérivées partielles dans le cadre de l'utilisation de la formule de Taylor.

4.4 Résolution d'un système d'équation en utilisant l'analyse par intervalles

Maintenant que nous voyons quels pouvaient être les différents moyens d'évaluer une fonction, il nous faut présenter de manière succincte comment résoudre des systèmes d'équations en utilisant l'analyse par intervalle. Nous avons déjà donné une piste dans la section précédente, lorsque nous avons abordé l'effet enveloppant. En effet avons fait remarquer qu'un des moyens de minimiser l'impact de l'effet enveloppant sur un intervalle consistait à le découper en sous-intervalles d'une taille arbitrairement petite. Cette idée rejoint d'un certain point de vue celle de la discrétisation classique utilisée en analyse numérique. Cependant compte tenu de la nature conservative des intervalles et contrairement à l'analyse numérique, nous ne risquons pas de perdre des solutions !

4.4.1 Les différents types de boîtes

Nous allons dans cette section présenter les différentes boîtes que l'on peut rencontrer au cours de la résolution d'un système d'équations intervalle. Soit $D \in \mathbb{R}^n$, l'ensemble des solution d'un système, et l'on va chercher à caractériser la position d'un boîte \mathbf{x}^I par rapport à ce domaine D .

Définition : Soit D un sous-ensemble de \mathbb{R}^n de dimension n .

- Une boîte est intérieure si : $\forall \mathbf{x} \in \mathbf{x}^I, x \in D$.

- Une boîte est extérieure si : $\nexists \mathbf{x} \in \mathbf{x}^I$ tel que $x \in D$. Cela signifie simplement $\mathbf{x}^I \cap D = \emptyset$.
- Une boîte est frontière quand elle n'est ni intérieure, ni extérieure.

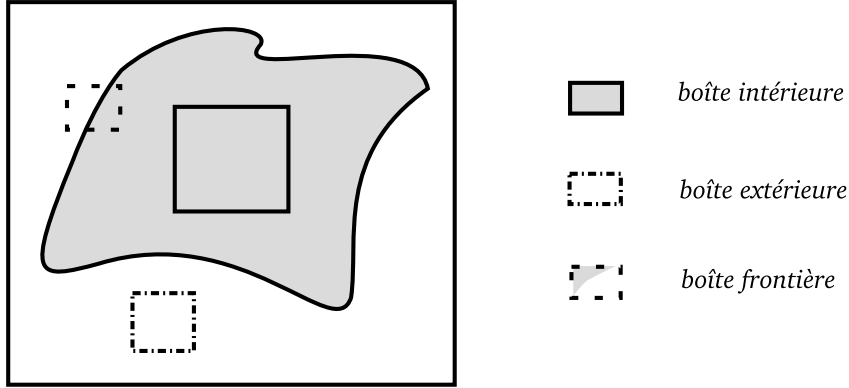


FIG. 4.3: Les différents types de boîtes pour un exemple où $D \in \mathbb{R}^2$ est grisé

Remarque : Dans un contexte numérique on distinguera les *boîtes frontières de taille atomique*, c'est à dire des boîtes frontière dont le diamètre est inférieur à un seuil fixé par l'utilisateur, des autres boîtes frontières.

Dans le cadre d'un système de dimension 0 (les solutions du systèmes sont des points), nous ne pouvons évidemment pas parler de boîte intérieure et nous parlons donc de boîte solution. Nous distinguons parmi ces dernières les boîtes solutions certifiées, qui sont de taille atomique, et pour lesquelles il a été prouvé qu'elles n'incluaient qu'une seule solution, solution qui doit être calculable avec une précision arbitraire par un algorithme numérique.

4.4.2 L'algorithme classique de résolution : Evaluation-Bisection

Soit le système d'équations $F(\mathbf{x}) = 0$ (où F est un vecteur de fonctions : $(f_1, ..f_n)$), que nous allons résoudre en utilisant l'analyse par intervalles, en recherchant les solutions incluses dans une boîte donnée \mathbf{x}^I qui va définir le *domaine de recherche*.

Le problème consiste donc à trouver l'ensemble $\{\mathbf{x} \in \mathbb{R}^n : F(\mathbf{x}) = 0 \text{ avec } \mathbf{x} \in \mathbf{x}^I\}$

Pour mener à bien la résolution de ce problème dans l'espace de recherche initial \mathbf{x}^I , nous introduisons un paramètre supplémentaire $\epsilon > 0$, qui va définir une précision. La solution d'un système d'équation intervalle retournée par l'algorithme sera une liste de boîtes de diamètre inférieur ou égal à ϵ (les boîtes sont alors de taille atomique).

Le principe de l'algorithme est très simple :

- **Evaluation :** Nous évaluons la fonction F sur la boîte \mathbf{x}^I , et s'il existe une fonction d'évaluation f_i^I telle que sur la boîte traitée $0 \notin f_i^I(\mathbf{x}^I)$, alors nous pouvons certifier qu'il n'existe pas de solution sur cette boîte. On parle de *test d'existence*.
- **Bisection :** Au contraire si $0 \in f_i^I(\mathbf{x}^I)$, alors nous créons deux nouvelles boîtes dont l'union est \mathbf{x}^I en coupant une composante de X en deux parties, généralement au

milieu. Cette opération se nomme la *bissection*.

nous utilisons ce principe de manière récursive, la bissection conduisant à alimenter une liste de boîtes dont le diamètre va décroissant. Certaines de ces boîtes vont être éliminées de la liste lors de l'évaluation et comme nous l'avons dit précédemment, nous nous arrêtons de bissecter une boîte quand celle-ci devient de taille atomique. L'algorithme s'arrête lorsque l'ensemble des boîtes de la liste ont été traitées. Compte tenu de la nature ensembliste des intervalles, on pourra certifier qu'on ne perd aucune solution.

Remarque : Par souci pédagogique, on a fait débiter l'algorithme avec une seule boîte initiale. Cependant, il peut tout aussi bien débiter avec une liste de boîtes.

4.4.3 Méthodes de filtrage

L'idée principale de l'algorithme précédent consiste à traiter, ou plutôt à exclure, dès que possible une partie du domaine initial représenté par une boîte ou une liste de boîtes. La boîte courante sera donc soit exclue dans sa globalité, soit bissectée. L'opération de bissection permet donc de raffiner l'évaluation de la boîte courante en la découpant en sous-parties.

Afin d'améliorer l'efficacité de l'algorithme d'évaluation-bissection, des *méthodes de filtrage* ont été développées. Leur but est de permettre parfois d'exclure la totalité d'une boîte en montrant qu'elle ne peut contenir de solution ou alors de réduire sa taille en éliminant des composantes qui ne contiennent pas de solution.

Ainsi si nous ajoutons des méthodes de filtrages à l'algorithme d'évaluation-bissection, il devient alors un algorithme de *contraction-bissection* puisqu'à chaque étape nous ne nous contentons plus d'évaluer le système étudié sur la boîte courante mais nous cherchons, également (via les méthodes de filtrages) à diminuer la taille de la boîte traitée.

Nous allons mieux comprendre l'utilisation du terme *contraction* en abordant deux méthodes de filtrage parmi les plus courantes que nous allons illustrer à l'aide d'exemples élémentaires pour faciliter la compréhension du lecteur. Nous renverrons ce dernier aux articles [2], [31] s'il souhaite approfondir ses connaissances sur les méthodes de filtrage.

4.4.3.1 La méthode de contraction 2B

L'algorithme 2B est un algorithme utilisé pour un grand nombre de problèmes intervalles et il est présent sur la plupart des outils de résolution de systèmes reposant sur l'arithmétique intervalle [34]. La procédure consiste à vérifier la consistance de l'équation par rapport aux domaines des différentes variables. Nous allons définir différentes fonctions, à partir d'une équation ou d'un système d'équations (qui se nomme fonctions de projection en programmation par contrainte), qui peuvent conduire à contracter le domaine initial voire réduire ce domaine à l'ensemble vide, ce qui revient à dire qu'il n'existe pas de solutions dans le domaine étudié. illustrons ce type de filtrage sur un exemple élémentaire afin de mieux en comprendre le principe.

Exemple : Nous cherchons à résoudre : $f(x) = 0$ tel que $f(x) = x^2 + 2x + 1$ avec $x \in [0, 10]$
Définissons deux fonctions de projection dans le cas présent qui vont isoler la variable x

et vont permettre de réduire le domaine initial. Pour cela nous avons réécrit l'équation $f(x) = 0$ sous deux formes mathématiquement équivalentes :

$$\begin{aligned} \bullet f_1(x) &: \frac{-x^2 - 1}{2} = x \\ \bullet f_2(x) &: \sqrt{(-2x - 1)} = x \text{ car } x \geq 0 \end{aligned}$$

Nous avons en fait réécrit le système de différentes manières afin de vérifier sa consistance, c'est à dire que si $f(x) = 0$ alors la variable x , qui à priori est dans le domaine $X = [0, 10]$, doit se trouver dans le domaine X_{new} défini par :

$$X_{new} = X_1 \cap X_2 \text{ avec } X_1 = f_1^I(X) \cap X \text{ et } X_2 = f_2^I(X) \cap X$$

Dans notre cas, on obtient :

$$X_1 = \left[-\frac{101}{2}, -\frac{1}{2}\right] \cap [0, 10] = \emptyset, \text{ ce qui signifie qu'il n'existe pas de solution au problème } f(x) = 0 \text{ pour } x \in [0, 10].$$

Pour X_2 , nous trouvons également l'ensemble vide car la fonction racine carrée n'est pas définie pour des valeurs négatives mais la première étape suffit puisque nous avons déjà prouvé qu'il n'existait pas de solution au système sur le domaine étudié.

4.4.3.2 La méthode de filtrage reposant sur le *rognage* ou *shaving*

Dans un premier temps illustrons le principe de cette méthode de filtrage pour le cas unidimensionnel, nous verrons ensuite qu'il n'existe pas de différences fondamentales quand nous passons au cas multidimensionnel 4.4.

L'idée principale du rognage consiste à exclure petit à petit du domaine étudié des zones sans solution aux extrémités du domaine. Nous cherchons à résoudre l'équation $f(x) = 0$ sur le domaine X . Commençons donc à rogner l'intervalle en partant par exemple de sa borne inférieure et tenter de rejoindre si cela est possible la borne supérieure.

Etape 1 : évaluons la fonction sur l'intervalle $[\underline{x}, \underline{x} + \epsilon]$ où ϵ est un critère fixé par l'utilisateur. Si après ce processus d'évaluation on a prouvé qu'il n'existe pas de solution sur ce domaine, alors on a commencé à rogner le bord inférieur, puisque les solutions éventuelles sont dans $[\underline{x} + \epsilon, \bar{x}]$.

Etape 2 : si nous sommes parvenu à exclure la portion $[\underline{x}, \underline{x} + \epsilon]$, alors nous continuons le même processus en doublant éventuellement le critère ϵ . Ainsi, nous évaluerons la fonction sur $[\underline{x} + \epsilon, \underline{x} + 3\epsilon]$.

Etape 3 : si au contraire, nous ne parvenons pas à éliminer une "petite" portion, alors on continue le processus de rognage en partant de la borne supérieure pour tenter de rejoindre la borne inférieure.

Remarques :

- Nous avons donc à la fin d'un processus de rognage trois résultats possibles : le domaine

initial n'est pas filtré, le domaine initial est partiellement réduit et enfin le dernier résultat implique qu'on a traité entièrement le domaine initial.

• Nous trouverons dans la littérature la dénomination $3B$ pour une méthode de filtrage du type *shaving* reposant sur la méthode de filtrage 2B et non pas sur la simple évaluation comme nous l'avons décrit ci-dessus.

Pour le cas multi-variables, nous effectuons le processus décrit précédemment sur chaque variable. Ainsi si nous cherchons à résoudre $f(x_1, \dots, x_n) = 0$ sur la boîte $\mathbf{x}^I = (X_1, \dots, X_n)$ nous débuterons la méthode de filtrage par exemple pour la variable associée au domaine X_1 . Dans un premier temps nous évaluerons la fonction sur $\mathbf{x}_0^I = (x_1, x_1 + \epsilon, X_2, \dots, X_n)$. A la fin du processus de rognage appliquée sur cette variable nous obtiendrons par exemple la boîte $\mathbf{x}_{\text{var1}}^I = (X'_1, X_2, \dots, X_n)$. Nous continuerons ensuite le processus pour toutes les autres variables.

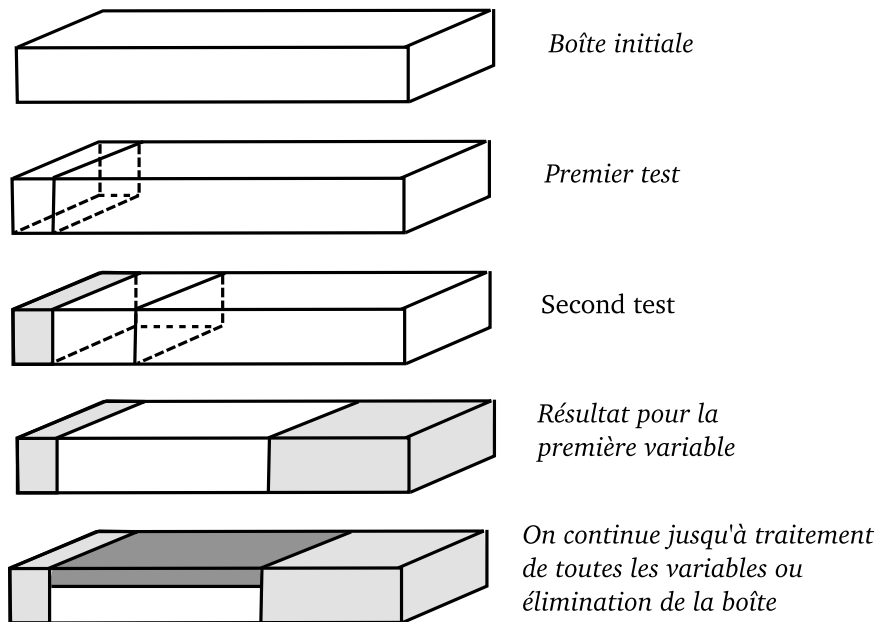


FIG. 4.4: Méthode de filtrage reposant sur le rognage, un test d'existence permet d'éliminer les boîtes grises

4.5 Résolution d'un système linéaire carré d'équations

Présentons de manière succincte ce qu'on entend par un système linéaire carré intervalle. La résolution de ce type de systèmes jouera un rôle important dans les prochains chapitres.

Soit le système linéaire carré : $A^I \mathbf{x}^I = \mathbf{b}^I$ avec A^I une matrice intervalle et \mathbf{x}^I et \mathbf{b}^I deux vecteurs intervalles.

La solution \mathcal{S} d'un tel système est défini par :

$$\mathcal{S}(A^I, \mathbf{b}^I) = \{x : A\mathbf{x} = \mathbf{b} \text{ avec } A \in A^I \text{ et } \mathbf{b} \in \mathbf{b}^I\} \quad (4.14)$$

\mathcal{S} sera bornée si A^I est régulière (c'est à dire que toutes les matrices scalaires A incluses dans A^I sont inversibles)

Cependant le calcul exact d'une telle solution est souvent très coûteux. En effet, pour un système de dimension n nous avons une solution ayant une forme d'étoile ou *papillon* avec au plus 2^n branches et le seul calcul de l'enveloppe convexe de \mathcal{S} se révèle être NP-dur. On se contente donc en général de calculer une boîte englobante de \mathcal{S} . Cela revient simplement à calculer les bornes maximales et minimales de la solution du système linéaire carré intervalle. Nous désignerons par $\sum(A^I, \mathbf{b}^I)$ l'enveloppe intervalle de la solution $\mathcal{S}(A^I, \mathbf{b}^I)$ (voir la figure 4.5).

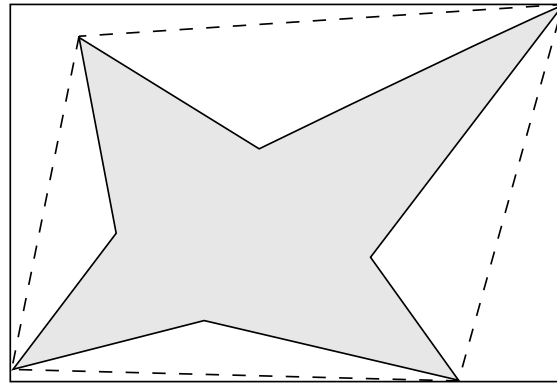


FIG. 4.5: En grisé la région solution d'un système linéaire intervalle, en pointillé une enveloppe convexe de cette région et en trait plein son approximation intervalle

4.5.1 Préconditionnement

Préconditionner un système $A\mathbf{x} = \mathbf{b}$ revient à le multiplier par une matrice constante M telle que $A\mathbf{x} = \mathbf{b} \iff MA\mathbf{x} = M\mathbf{b}$

En analyse par intervalle, un des choix possibles pour la matrice M qui sert de préconditionneur est la matrice milieu $A^c = \text{mid}(A^{-1})$ qui a pour but de rapprocher MA^I de la matrice identité, réduisant ainsi les dépendances dans le produit $MA^I\mathbf{x}^I$

En règle générale l'utilisation d'un préconditionneur favorise largement la résolution du système linéaire que nous allons nous attacher à décrire dans la prochaine section, mais il arrive que pour certains cas, comme par exemple les matrices à diagonales dominantes, la solution $\sum(A^I, \mathbf{b}^I)$ soit englobée dans $\sum(MA^I, M\mathbf{b}^I)$. Pour une analyse plus poussée du preconditionnement nous conseillerons [43] au lecteur.

4.5.2 Résolution

Présentons les principales méthodes de résolution de système linéaire carré utilisé en analyse par intervalles.

4.5.2.1 Le pivot de Gauss

Il est très simple d'adapter cet algorithme à l'analyse par intervalles. En effet il suffit simplement de remplacer les termes scalaires par les intervalles, mais cela peut conduire à une surestimation parfois importante de la solution. Le pivot de Gauss peut être vu comme une triangularisation de la matrice puis un algorithme de remontée pour la résolution. On peut également faire une factorisation LU et l'algorithme de résolution revient à un algorithme de remontée puis de descente.

Remarquons que si la matrice A^I et le vecteur \mathbf{b}^I sont tous les deux des intervalles dégénérés alors le vecteur \mathbf{x}^I donne des bornes sur les erreurs d'arrondis liés à cette méthode de résolution.

Un des problèmes majeur du pivot de Gauss est la surestimation du résultat auquel il peut mener sauf cas spécial (matrice à diagonale dominante entre autres [43]). Aussi il est vivement conseillé d'utiliser un préconditionneur comme la matrice $\text{mid}(A^{-1})$ par exemple. Les résultats après conditionnement sont en règle générale bien plus précis.

4.5.2.2 La méthode Gauss-Seidel

Comme pour la méthode du pivot de Gauss il est intéressant de raisonner avec un système préconditionné.

Soit donc le système : $M^I \mathbf{x}^I = \mathbf{r}^I$

On va regarder ce qui se passe pour la i -ème composante du vecteur intervalle r^I . D'un point de vue symbolique on peut écrire la relation :

$$M_{i1}x_1 + \dots + M_{in}x_n = r_i \text{ qui peut-être résolue pour } x_i \text{ (sous réserve que } M_{ii} \neq 0)$$

$$x_i = (r_i - M_{i1}x_1 + \dots + M_{i,i-1}x_{i-1} + M_{i,i+1}x_{i+1} + \dots + M_{in}x_n)/M_{ii}$$

Si l'on suppose que des bornes X_i sont connues pour les composantes x_i on pourra obtenir de nouvelles bornes Y_i sur \mathbf{x}^I . On va donc les calculer en respectant la condition $0 \notin [M_{ii}]$.

$$Y_i = (R_i - [M_{i1}]X'_1 + \dots + [M_{i,i-1}]X'_{i-1} + [M_{i,i+1}]X_{i+1} + \dots + [M_{in}]X_n)/[M_{ii}] \quad (4.15)$$

avec $X'_i = X_i \cap Y_i$.

Il arrive même parfois que Gauss-Seidel avec $0 \in [M_{ii}]$ réduise la taille du vecteur intervalle initial \mathbf{x}^I si l'on utilise le principe de la division étendue. Si jamais l'intersection $X'_i \cap X_i = \emptyset$ alors le vecteur intervalle initial ne contient pas de solution. Il peut également arriver que Gauss-Seidel n'améliore pas les bornes dans certains cas (estimé initial déjà compris dans l'ensemble des solutions). Nous pouvons également voir la méthode de Gauss-Seidel comme une méthode de filtrage du type 2B sur les variables liées à \mathbf{x}^I . De plus le caractère itératif de la méthode Gauss-Seidel peut-être utilisé avec

un certain avantage dans le cadre d'un algorithme de type bisection-évaluation. On stockera le résultat à chaque étape et il sera réutilisé comme condition initiale à l'étape suivante.

4.6 Transition

Après cette introduction succincte à l'analyse par intervalles, nous avons désormais les moyens de nous attaquer à la détermination de l'espace de travail statique pour un ensemble de forces et d'orientations données. Dans le chapitre suivant, nous allons nous attacher à expliquer comment calculer une approximation de l'intégralité de l'espace de travail statique et nous décrirons les principes d'un algorithme permettant d'arriver à nos fins. De plus, l'analyse par intervalles nous permettra également d'introduire des incertitudes sur les positions des points d'attache des jambes du robot sur la base ou sur la plate-forme, ce qui correspond soit aux tolérances de fabrication, soit à une famille de robots contenant de manière sûre le robot réel. Notre algorithme d'approximation de l'espace de travail statique fournira alors une zone approchant l'intersection de l'espace de travail statique de chacun des robots de la famille, zone qui sera forcément incluse dans l'espace de travail statique du robot réel. Nous obtiendrons donc un ensemble de poses de la plate-forme mobile pour lesquelles nous serons en mesure de certifier qu'elles sont sûres au sens statique, c'est-à-dire que le mécanisme réel ne risque pas de subir de détériorations dues aux forces/couples subis par ses jambes.

APPROXIMATION DE L'ESPACE DE TRAVAIL STATIQUE PAR L'ANALYSE PAR INTERVALLES

5.1 Introduction

Dans les chapitres précédents nous avons proposé un algorithme de calcul de coupes planes de l'espace de travail statique pour un robot soumis à une charge fixée \mathcal{F} . Nous avons introduit les éléments de base de l'analyse par intervalles. Clairement, il est intéressant pour nous de :

- calculer l'intégralité de W_s et non plus des sections planes.
- calculer W_s non plus pour une charge mais un ensemble de charges.

La généralisation de la méthode précédemment décrite à ces cas semble très difficile. Dans ce chapitre, nous allons montrer que l'analyse par intervalles autorise, sous conditions, ces généralisations. Cela va nous permettre de déterminer indifféremment l'espace de travail statique de robots parallèles avec un unique mode d'assemblage. Nous verrons que cette approche a un autre avantage majeur : elle permet de prendre en compte des incertitudes sur la géométrie du robot, l'espace de travail statique alors obtenu sera ainsi valide *même pour le robot réel*.

Définition : Nous considérons donc d'un point de vue intervalle, l'espace de travail statique comme l'ensemble des poses χ de la plate-forme mobile vérifiant quel que soit l'effort extérieur $\mathcal{F} \in \mathcal{F}^{\mathbf{I}}$ que les forces τ appliquées par les différentes jambes vérifient $\|\tau\|_{\infty} \leq \tau_{max}$. $\|\cdot\|_{\infty}$ désigne la norme infinie, c'est-à-dire : $\|\tau\|_{\infty} = \text{Max}(|\tau_i|)$. Formellement

$$W_s = \{\chi \in W : \forall \mathcal{F} \in \mathcal{F}^{\mathbf{I}} \|\tau\|_{\infty} \leq \tau_{max}, \text{ et vérifie } \mathbf{J}^{-\mathbf{T}}(\chi)\tau = \mathcal{F}\} \quad (5.1)$$

5.2 Objectifs

Le calcul de W_s ainsi défini est évidemment très complexe à obtenir. Notre objectif va donc être d'en obtenir une *approximation*, tout en restant maître de la *qualité* de cette approximation. En effet, nous allons pouvoir la quantifier.

Le résultat R obtenu par notre calcul va vérifier les propriétés suivantes :

- 1) R est inclus dans l'espace de travail statique de manière certifiée.

- 2) nous pouvons calculer le volume V de R .
- 3) un majorant V_E du volume de W_s sera obtenu.
- 4) il sera possible de choisir un $\mu > 0$ arbitraire tel que $V_E - V < \mu$. Toutefois selon la valeur de μ le temps de calcul pourra être important.

Plus précisément, le résultat obtenu sera constitué d'une liste de boîtes, dont la dimension correspondra au nombre de degrés de liberté du robot. Chacune de ces boîtes est strictement incluse dans l'espace de travail statique du mécanisme. La nature même du résultat garantit les propriétés 1) et 2).

Un élément essentiel de notre algorithme va être la capacité à déterminer si une boîte est :

- totalement à l'intérieur de W_s (boîte intérieure).
- totalement à l'extérieur de W_s (boîte extérieure).
- possiblement avec une partie à l'intérieur et l'autre à l'extérieur (boîte frontière).

5.3 Boîte et espace de travail statique

Ainsi un *élément clé* de notre algorithme repose sur notre capacité à déterminer, si une boîte \mathbf{x}^I (représentant un ensemble de poses χ du mécanisme étudié) est entièrement incluse dans l'espace de travail statique ou au contraire si elle est exclue de ce dernier.

Une boîte est incluse dans l'espace de travail statique si nous avons :

$$\forall \chi \in \mathbf{x}^I, \forall \mathcal{F} \in \mathcal{F}^I, \exists \tau \in \tau_{\max}^I : J^{-T}(\chi)\tau = \mathcal{F} \quad (5.2)$$

Réciproquement, une boîte est à l'extérieur de W_s si :

$$\forall \chi \in \mathbf{x}^I, \forall \tau \in \tau_{\max}^I, \exists \mathcal{F} \in \mathcal{F}^I : J^{-T}(\chi)\tau \neq \mathcal{F} \quad (5.3)$$

Précisons que τ_{\max}^I est une boîte dont chaque composante est l'intervalle $[-\tau_{\max}, \tau_{\max}]$. Enfin une boîte qui ne vérifie ni l'équation (5.2) ni l'équation (5.3) sera soumise à une bisection.

5.3.1 Test d'inclusion

Par une boîte \mathbf{x}^I nous désignons un ensemble de poses χ de la plate-forme mobile. L'évaluation de la transposée de la jacobienne cinématique inverse nous conduit à une matrice intervalle $\mathbf{A}^I = (\mathbf{J}^{-T}(\mathbf{x}^I))^I$ telle que :

- 1) $\mathbf{J}^{-T}(\mathbf{x}^I) \subseteq \mathbf{A}^I$
- 2) il pourra exister des matrices $A \in \mathbf{A}^I$ telles que $A \notin \mathbf{J}^{-T}(\mathbf{x}^I)$

Une condition suffisante (mais pas nécessaire car $\mathbf{J}^{-T}(\mathbf{x}^I) \subseteq \mathbf{A}^I$) pour qu'une boîte \mathbf{x}^I soit incluse dans l'espace de travail statique peut être établie sous la forme :

$$\forall A \in \mathbf{A}^I, \forall \mathcal{F} \in \mathcal{F}^I, \exists \tau \in \tau_{\max}^I : A\tau = \mathcal{F} \quad (5.4)$$

Grâce à cette condition suffisante, nous allons pouvoir définir un test d'inclusion pour une boîte qui implique la résolution du système linéaire par intervalle $\mathbf{A}^{\mathbf{I}}\tau_{\text{sol}} = \mathcal{F}^{\mathbf{I}}$. Rappelons que ce genre de problème a déjà été étudié et que l'ensemble solution d'un tel système est défini par :

$$\sum(\mathbf{A}^{\mathbf{I}}, \mathcal{F}^{\mathbf{I}}) = \{\tau_{\text{sol}} : \exists A \in \mathbf{A}^{\mathbf{I}}, \exists \mathcal{F} \in \mathcal{F}^{\mathbf{I}}, A\tau_{\text{sol}} = \mathcal{F}\} \quad (5.5)$$

Propriété : Si $\sum(\mathbf{A}^{\mathbf{I}}, \mathcal{F}^{\mathbf{I}}) = \tau_{\text{sol}}^{\mathbf{I}}$ est strictement inclus dans $\tau_{\text{max}}^{\mathbf{I}}$, alors la boîte $\mathbf{x}^{\mathbf{I}}$ qui génère $\mathbf{A}^{\mathbf{I}}$ fait partie de l'espace de travail statique.

Le test d'inclusion pour une boîte $\mathbf{x}^{\mathbf{I}}$ consistera donc à calculer l'ensemble des matrices $\mathbf{A}^{\mathbf{I}} = (\mathbf{J}_{\text{inv}}^{\mathbf{I}})^T(\mathbf{x}^{\mathbf{I}})$ puis à résoudre le système linéaire $\mathbf{A}^{\mathbf{I}}\tau_{\text{sol}} = \mathcal{F}^{\mathbf{I}}$ et enfin à vérifier que l'ensemble solution $\tau_{\text{sol}}^{\mathbf{I}}$ est bien strictement inclus dans $\tau_{\text{max}}^{\mathbf{I}}$. Bien entendu, nous pourrions faire appel à toutes les techniques exposées dans le chapitre 4, qui nous permettront d'obtenir le plus petit encadrement possible de $\tau_{\text{sol}}^{\mathbf{I}}$ (par exemple, en préconditionnant le système $\mathbf{A}^{\mathbf{I}}\tau_{\text{sol}} = \mathcal{F}^{\mathbf{I}}$).

5.3.2 Tests d'exclusion

5.3.2.1 Tests par résolution du système linéaire intervalle

Avec le même procédé que le test d'inclusion, on définit un test d'exclusion en utilisant le résultat de la résolution du système linéaire par intervalle $\mathbf{A}^{\mathbf{I}}\tau_{\text{sol}} = \mathcal{F}^{\mathbf{I}}$. En effet, si $\tau_{\text{sol}}^{\mathbf{I}} \cap \tau_{\text{max}}^{\mathbf{I}} = \emptyset$, nous pouvons conclure que :

$$\forall \chi \in \mathbf{x}^{\mathbf{I}}, \forall \mathcal{F} \in \mathcal{F}^{\mathbf{I}}, \nexists \tau \in \tau_{\text{max}}^{\mathbf{I}} : J^{-T}(\chi)\tau = \mathcal{F} \quad (5.6)$$

ce qui implique que la relation (5.3) est vérifiée et donc la boîte $\mathbf{x}^{\mathbf{I}}$ peut-être exclue de l'espace de travail statique puisque pour toute pose $\exists i$ tel que $|\tau_i| > \tau_{\text{max}}$.

Notons que dans ce cas il n'est même pas nécessaire de considérer $\mathcal{F}^{\mathbf{I}}$ dans son intégralité : si la relation (5.3) est vérifiée ne serait-ce que pour un seul $\mathcal{F} \in \mathcal{F}^{\mathbf{I}}$ alors la boîte peut-être exclue.

5.3.2.2 Test par évaluation du torseur statique

Nous pouvons également définir un autre test d'exclusion très simple. Soit :

$$\mathbf{b}^{\mathbf{I}} = \mathbf{A}^{\mathbf{I}}\tau_{\text{max}}^{\mathbf{I}} \quad (5.7)$$

Il consiste à multiplier la matrice intervalle $\mathbf{A}^{\mathbf{I}}$ obtenue après évaluation par le vecteur intervalle $\tau_{\text{max}}^{\mathbf{I}}$.

$\mathbf{b}^{\mathbf{I}}$ constitue alors une sur-approximation de l'ensemble des vecteurs qui peuvent être générés par l'ensemble des matrices $A \in \mathbf{A}^{\mathbf{I}}$ et l'ensemble des vecteurs $\tau \in \tau_{\text{max}}^{\mathbf{I}}$. Par conséquent, si la relation $\mathcal{F}^{\mathbf{I}} \subseteq \mathbf{b}^{\mathbf{I}}$ n'est pas satisfaite, alors il existe au moins un des \mathcal{F} de $\mathcal{F}^{\mathbf{I}}$ qui ne peut pas être généré par les matrices de $\mathbf{A}^{\mathbf{I}}$ et les vecteurs de $\tau_{\text{max}}^{\mathbf{I}}$ ce qui revient bien à vérifier (5.3).

5.3.2.3 Test de Rohn

Nous pouvons définir un autre test d'exclusion, en utilisant le travail de Rohn [25], d'ailleurs exploité par Gouttefarde [32]. Rappelons qu'une boîte est pleinement incluse dans l'espace de travail statique si la propriété 5.2 est vérifiée. Rohn a montré que cette propriété était équivalente à :

$$\forall A \in \mathbf{A}^{\mathbf{I}}, \forall \mathcal{F}_{\mathbf{y}} \in \{\mathcal{F}_{\mathbf{y}}\}, \exists \tau \in \tau_{\max}^{\mathbf{I}} : A\tau = \mathcal{F}_{\mathbf{y}} \quad (5.8)$$

où $\{\mathcal{F}_{\mathbf{y}}\}$ désigne l'ensemble des arêtes de la boîte $\mathcal{F}^{\mathbf{I}}$. Ainsi si (5.8) est faux, la boîte qui génère l'ensemble $\mathbf{A}^{\mathbf{I}}$ est extérieure à l'espace de travail statique ce qui va impliquer :

$$\exists \mathcal{F}_{\mathbf{y}} \in \{\mathcal{F}_{\mathbf{y}}\} : \forall \mathbf{x} \in \mathbf{x}^{\mathbf{I}}, \forall \tau \in \tau_{\max}^{\mathbf{I}}, A\tau \neq \mathcal{F}_{\mathbf{y}} \quad (5.9)$$

Cherchons à exclure des portions de $\mathbf{x}^{\mathbf{I}}$ et $\tau_{\max}^{\mathbf{I}}$ pour lesquelles le système est inconsistent, c'est à dire $A\tau \neq \mathcal{F}_{\mathbf{y}}$. Nous faisons alors appel à des techniques utilisées pour la résolution de système discrets [18] [19] [31]. Ces méthodes, comme la 2B, ne résolvent pas à proprement parler le système d'équations, mais elles réduisent la largeur des ensembles $\mathbf{x}^{\mathbf{I}}$ et $\tau_{\max}^{\mathbf{I}}$.

Si après application de ces heuristiques nous obtenons $\mathbf{x}^{\mathbf{I}} = \emptyset$ ou $\tau_{\max}^{\mathbf{I}} = \emptyset$ pour l'un des $\{\mathcal{F}_{\mathbf{y}}\}$ alors nous pouvons immédiatement considérer la boîte $\mathbf{x}^{\mathbf{I}}$ comme complètement extérieure à l'espace de travail statique. Contrairement aux tests précédents, ce test ne se limite pas à l'exclusion de la boîte $\mathbf{x}^{\mathbf{I}}$ entière, nous pouvons exclure des portions situées sur les bords de $\mathbf{x}^{\mathbf{I}}$ si les techniques de consistances ont prouvé qu'il n'y avait pas de solution dans ces portions. Cela est dû au fait que nous considérons \mathbf{x} comme une variable à part entière, nous ne nous contentons plus de tester la boîte mais nous cherchons également à la contracter si cela est possible.

5.4 Algorithme de calcul de l'espace de travail statique

Afin d'obtenir une approximation de l'espace de travail statique, nous nous fixons les objectifs suivants :

- 1) obtenir une liste de boîtes, de diamètre supérieur à ϵ , qui sont incluses dans l'espace de travail statique. Elles sont stockées dans un fichier *IN*.
- 2) obtenir une liste de boîtes, de diamètre inférieur à ϵ , qui n'ont pas pu être exclues de W_s . Elles sont stockées dans le fichier *BORDER*.
- 3) créer éventuellement une liste de boîtes, de diamètre supérieur à ϵ , qui n'appartiennent pas à W_s et qui seront stockées dans un fichier *OUT*.

Pour atteindre ces objectifs, nous disposons des moyens suivant :

- a) un test d'inclusion.
- b) un test d'exclusion.
- c) une liste L de boîtes B_i (avec $i = 1..m$) dont l'union correspond à l'ensemble des poses que nous souhaitons étudier.

d) un moyen de gérer la liste : la parcourir, la modifier, ajouter ou supprimer des éléments.

l'algorithme est décrit dans la figure 5.1, il débute avec $i = 1$ comme index initial et m le nombre de boîtes dans la liste L

Etape 1 : si $m = 0$ arrêt de l'algorithme.

Etape 2 : B_i soumise au test d'exclusion. Si B_i est exclue, nous la stockons dans OUT . Retour à l'étape 1 avec $i = i + 1$ et $m = m - 1$ Sinon nous passons à l'étape suivante.

Etape 3 : B_i soumise au test d'inclusion. Si B_i est incluse, nous la stockons dans IN . Retour à l'étape 1 avec $i = i + 1$ et $m = m - 1$. Sinon nous passons à l'étape suivante.

Etape 4 : Si $Diam(B_i) < \epsilon$, stockons B_i dans $BORDER$. Retour à l'étape 1 avec $i = i + 1$ et $m = m - 1$. Sinon bisection de la boîte B_i , stockage des deux boîtes obtenues dans L , retour à l'étape 1 avec $i = i + 1$ et $m = m + 1$.

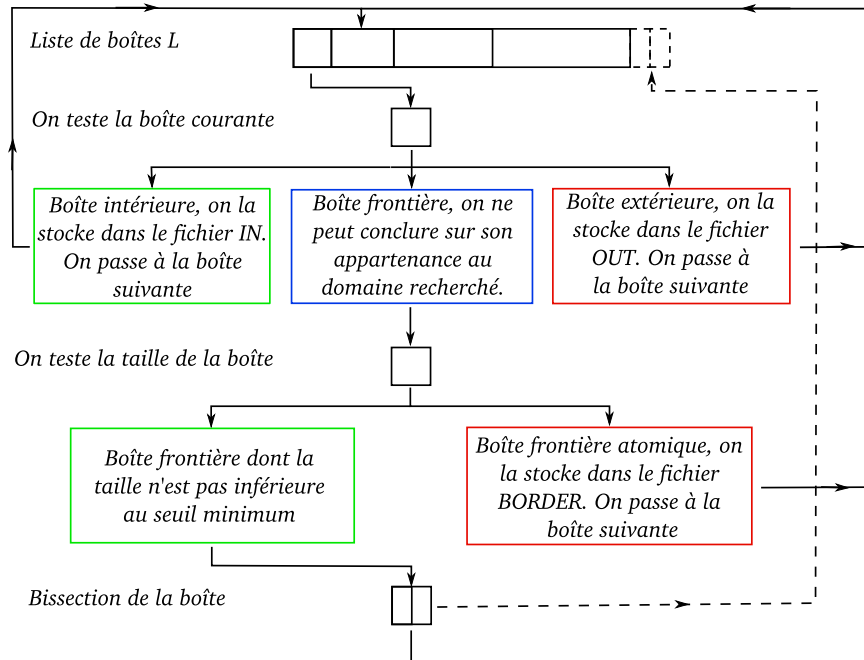


FIG. 5.1: algorithme "Branch and Prune"

5.4.1 Gestion de la bisection

La bisection se décompose en deux étapes :

- 1) choisir une variable à bissecter.
- 2) couper la boîte selon la variable.

Le choix de la variable à bissecter peut avoir une grande influence sur la qualité de R et/ou la vitesse de l'algorithme. Il existe différentes méthodes pour effectuer ce choix, nous allons en citer trois :

- méthode 1 : les variables sont coupées à tour de rôle (*round robin*).
- méthode 2 : la variable qui a l'intervalle le plus large est coupée.
- méthode 3 : la variable qui a le plus "*d'influence*" sur le résultat est coupée.

La méthode 1 a comme but de ne favoriser aucune variable, alors que la méthode 2 permet d'obtenir le plus rapidement possible des boîtes dont le diamètre est inférieur au diamètre minimal donc des boîtes frontières. Mais ces deux méthodes ne prennent pas en compte l'influence des variables sur le problème. La mesure de cette influence est délicate. Cependant mentionnons la fonction SMEAR de Kearfott [27] qui utilise des dérivées pondérées.

Pour couper une boîte $B = (X_1, \dots, X_n)$ selon la i -ème variable il suffit de couper le i -ème intervalle X_i en son milieu. Nous obtenons alors deux nouvelles boîtes qu'il faut ensuite stocker dans la liste L .

5.4.2 Gestion du Stockage

Pour stocker une boîte B de dimension n , nous avons besoin de $2n$ flottants, chaque flottant étant associé à une borne d'un intervalle. B_{l1} et B_{r1} sont les intervalles obtenus après la bisection de B_1 . Deux questions se posent :

Question 1 : laquelle des boîtes B_{l1} ou B_{r1} doit être traitée en priorité ?

Question 2 : en supposant que l'une des boîtes B_{l1} ou B_{r1} va occuper la place mémoire de B_1 , où doit-on placer la boîte restante ? Au début ou à la fin de la liste L ?

Si notre problème consiste à traiter tout l'ensemble X , défini par l'union des boîtes de la liste L , afin d'obtenir une approximation de l'espace de travail statique, traiter B_{l1} ou B_{r1} en priorité pourra influencer la taille du stockage nécessaire mais n'aura aucune influence sur la vitesse d'exécution de notre algorithme. En effet l'algorithme s'arrête quand il ne reste plus de boîtes à traiter dans L . Par contre si nous cherchons à certifier qu'un ensemble X donné est inclus dans l'espace de travail statique le choix peut avoir son importance. En effet si X n'est pas inclus dans W_s , alors l'algorithme s'arrêtera dès que nous rencontrons la première boîte *BORDER* ou *OUT*. Parmi B_{l1} et B_{r1} , il est donc important de considérer l'élément qui a la plus grande probabilité d'être exclu. Certains moyens permettant d'estimer cette probabilité seront abordés ultérieurement.

Avant de répondre à la seconde question, précisons pour quelle raison l'une des boîtes obtenues après bisection doit occuper l'emplacement mémoire de B_1 : B_1 ayant déjà été traitée, il est inutile de continuer à la stocker. Au lieu d'ajouter deux nouvelles boîtes à la liste, autant utiliser la place mémoire de B_1 pour stocker B_{l1} ou B_{r1} . Si l'intervalle restant est placé en début de liste (respectivement en fin de liste), nous parlons de *parcours en profondeur* (respectivement de *parcours en largeur*). La résolution de notre problème d'analyse par intervalle peut être vu comme trouver les feuilles *IN*, *OUT* et *BORDER* d'un arbre (figure 5.2). Tandis que le parcours en largeur peut conduire, au cours du processus, à passer d'une branche à une autre (passage entre les boîtes dont les index de droite sur la figure 5.2 sont 4 et 5), le parcours en profondeur explore toutes les feuilles d'une même branche avant de passer à une autre branche.

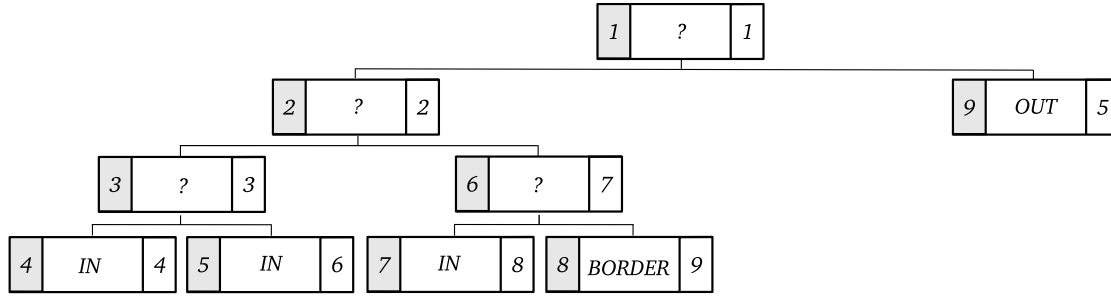


FIG. 5.2: Arbre correspondant à la résolution d'un problème numérique d'analyse par intervalles. Au centre des feuilles de l'arbre, le résultat du traitement de la boîte, à gauche en grisé l'ordre de traitement de la boîte avec un parcours en profondeur et à droite l'ordre de passage de la boîte avec un parcours en largeur.

En règle générale, le parcours en largeur amène à une taille de stockage plus importante de boîtes au cours du processus de résolution. Cependant si nous cherchons à certifier que l'ensemble X est bien inclus dans l'espace de travail statique, nous pouvons obtenir un résultat négatif plus rapidement. Si l'espace de stockage est limitant, il est également possible que nous obtenions un résultat avec un parcours en profondeur alors que le parcours en largeur s'arrête faute d'espace. Il est toutefois possible d'établir une méthode mixte qui va passer du parcours en largeur au parcours en profondeur si un nombre donné d'intervalles a été atteint.

5.4.3 Qualité de l'approximation de W_s

Maintenant que l'algorithme a été complètement décrit, intéressons nous à la qualité de l'approximation de W_s . Nous possédons une liste de boîtes IN dont le volume total est V_{IN} et une liste de boîtes $BORDER$ dont le volume est V_{BORDER} . L'ensemble des boîtes $BORDER$ et IN constitue un majorant de l'espace de travail statique. Ainsi un majorant V_E du volume de l'espace de travail statique sera obtenu par $V_E = V_{IN} + V_{BORDER}$.

Nous définissons alors un indice de qualité $I_{quality} > 0$ de l'approximation de W_s par :

$$I_{quality} = \frac{V_{BORDER}}{V_{IN} + V_{BORDER}} \quad (5.10)$$

Nous avons donc atteint les 4 objectifs que nous nous étions fixés précédemment. Il est alors possible de définir un μ maximal comme un critère à atteindre : si l'on obtient $I_{quality} < \mu$ alors nous pouvons considérer notre objectif d'approximer W_s comme atteint. Il va de soit que plus μ est petit, plus le temps de calcul va être important. Maintenant expliquons comment l'aspect incrémental de notre algorithme peut permettre d'améliorer la qualité de l'approximation de W_s .

5.4.4 Aspect incrémental et amélioration de la qualité

Notre algorithme nous fournit une liste de boîtes $BORDER$ dont le diamètre est inférieur à ϵ . Ces boîtes ont été stockées dans un fichier. Si nous exécutons à nouveau notre

algorithme en partant uniquement de ces boîtes frontières en remplaçant ϵ par $\frac{\epsilon}{2}$ cela peut amener à :

- obtenir de nouvelles boîtes intérieure, donc V_{IN} augmente.
- obtenir de nouvelles boîtes extérieure donc V_{BORDER} diminue.

Notre approximation de W_s va donc être de meilleur qualité. Le choix de ϵ pour la première itération est délicat car il est difficile de savoir à priori quelle valeur conduira à un résultat à la fois non vide et à un temps de calcul faible. L'aspect incrémental permet de choisir une bonne stratégie qui va être de démarrer avec un ϵ "grand" puis de faire décroître automatiquement ϵ pour atteindre la qualité que nous nous sommes fixés comme objectif à atteindre.

5.4.5 Implantation distribuée

La résolution numérique de problèmes d'analyse par intervalles consiste principalement à traiter une liste de boîtes. Comme le traitement de chaque boîte s'effectue de manière indépendante, il apparaît approprié d'utiliser une *implantation distribuée*. L'exécution simultanée sur différents processeurs du traitement des différentes boîtes accélère la résolution. On trouvera une implantation distribuée sur le logiciel ALIAS.

Une solution possible consiste à utiliser une méthode classique d'implantation distribuée : *maître-esclaves*. Le processeur maître gère, seul, la liste des boîtes à traiter. Dès qu'un des processeurs esclave est libre de toute tâche, le maître lui envoie une ou des boîtes à traiter. Les processeurs esclaves possèdent leur propre liste de boîtes et effectuent un certain nombre d'itérations de l'algorithme de résolution. Une fois un critère d'arrêt atteint (nombre maximum d'itérations atteint, toutes les boîtes assignées ont été traitées,...), le processeur *esclave* renvoie sa liste de boîtes et/ou signifie au processeur maître qu'il est libre de toute tâche. Le processeur maître lui assigne alors de nouvelles boîtes à traiter. Avec cette méthode la plupart des calculs sont effectués par les esclaves. Évidemment si tous les processeurs esclaves sont occupés, le processeur maître débute quelques itérations de l'algorithme tout en restant attentif à l'état de ses esclaves. Cette méthode est pratique pour la résolution de l'analyse par intervalles car :

- peu de données sont échangées entre le maître et les esclaves. Typiquement pour un robot à n degrés de liberté, nous avons n inconnues donc au cours d'un échange nous envoyons par boîte un maximum de $2n$ flottants. Nous pourrions être amené à augmenter la taille des messages (ajout de variable de contrôle, informations supplémentaires sur les boîtes) mais la taille des messages restera faible.
- dans notre cas, le traitement d'une boîte va généralement prendre bien plus de temps que l'envoi de boîtes.

Soit T le temps de calcul que prend un processeur unique pour fournir un résultat R et soit m le nombre de processeurs utilisés dans le calcul parallèle. Ces m processeurs fournissent R en un temps T_p . Nous pouvons mesurer l'efficacité du calcul parallèle par $\alpha = \frac{T}{T_p}$. Idéalement avec une mémoire partagée par tous les processeurs α serait égal à m mais la gestion des échanges conduit typiquement à un α de l'ordre de $0.8m$.

Un autre aspect intéressant de l'implantation distribuée consisterait à multiplier les approches de résolution sur les différents processeurs esclaves afin de trouver l'approche la plus efficace.

5.5 Éléments pour l'implantation de différentes stratégies de résolution

L'utilisation d'un test d'exclusion plutôt qu'un autre associé à une méthode de bisection spécifique définit une *stratégie de résolution*. Une stratégie se révèle plus ou moins performante suivant le ou les problèmes étudiés. Gardons à l'esprit qu'il n'existe pas une méthode de résolution parfaite pour notre problème d'analyse par intervalles. La modification d'un paramètre pouvant conduire à des résultats de piètre qualité et/ou à une faible vitesse de résolution. Il est donc important de multiplier les approches différentes car les écarts peuvent se révéler importants suivant la taille, le diamètre des boîtes, la nature des équations traitées. L'une des difficultés majeures des algorithmes numériques d'analyse par intervalles consiste à obtenir une stratégie adéquate pour un problème donné. Dans notre cas, outre la vitesse de résolution, la capacité à fournir une bonne qualité d'approximation est un élément fondamental. En vue de répondre à ces objectifs, nous allons détailler l'un des facteurs fondamental pour la définition d'une stratégie : l'évaluation.

5.5.1 L'évaluation

5.5.1.1 Différentes stratégies pour l'évaluation

L'évaluation est présente pour chaque itération de l'algorithme aussi son coût ne doit pas être négligé. On comprend que plus l'évaluation des coefficients de la matrice jacobienne est "*précise*" et plus les tests d'inclusion et d'exclusion risquent de renvoyer des résultats concluants. Nous avons déjà vu dans le chapitre 4 différentes méthodes possibles pour évaluer une fonction :

- 1) L'*évaluation naturelle* de la fonction, c'est à dire que nous remplaçons chaque variable par l'intervalle qui lui est associé
- 2) Utilisation de la formule de Taylor au premier ou au second ordre.
- 3) Utilisation de la monotonie.

Nous allons donner des principes généraux sur l'évaluation des coefficients de la matrice mais il faut bien comprendre qu'il peut exister certains cas où ils ne seront pas vérifiés. Plus la boîte est "*grande*" et plus la probabilité d'obtenir des boîtes extérieures ou intérieures est faible, même si l'évaluation des coefficients est très précise. Les tests d'inclusion ou d'exclusion risquent donc fort de renvoyer des résultats non concluants. Ainsi l'utilisation de la formule de Taylor, l'utilisation de la monotonie voire la combinaison de ces deux méthodes vont se révéler peu efficace en terme de temps de calcul comparé à l'évaluation naturelle de la fonction, puisque ces méthodes gourmandes en temps de calcul n'amèneront pas d'amélioration sensible de l'évaluation.

A l'inverse plus la boîte va être "*petite*" et plus nous avons de chance de conclure sur

l'appartenance ou non d'une boîte à W_s . Une évaluation plus précise peut alors conduire à améliorer les résultats des différents tests. Nous pourrions donc potentiellement exclure des boîtes de plus grande taille avec l'utilisation des dérivées. Nous pourrions également combiner la formule de Taylor et l'utilisation de la monotonie pour obtenir des évaluations plus précises des coefficients de la matrice et/ou des dérivées de ces coefficients comme cela est fait dans le logiciel *ALIAS*.

En conséquence une stratégie possible consiste à définir un seuil γ pour le diamètre des boîtes. Si le diamètre de la boîte courante est supérieur à γ , nous n'appliquerons que l'évaluation naturelle.

5.5.1.2 Formulation Matrice

Nous avons déjà insisté sur l'importance de la formulation d'un problème d'analyse par intervalles, afin de limiter les occurrences multiples. L'utilisation d'*ALIAS-Maple* nous permet déjà un pré-traitement symbolique qui empêche l'utilisation de mauvaise formulation de la matrice $J^{-T}(\chi)$. Cependant, il peut-être intéressant de modifier le système afin de travailler avec une nouvelle matrice $M^T(\chi)$ dont l'expression symbolique est plus "*simple*" que $J^{-T}(\chi)$. Comme nous l'avons déjà expliqué dans le chapitre 3, on obtient M^T à partir de J^{-T} en sortant les dénominateurs ρ_i qui apparaissent pour chaque colonne de J^{-T} . Nous pouvons donc écrire $M^T = J^{-T}D$ avec $D = \text{diag}(\frac{1}{\rho_1}, \dots, \frac{1}{\rho_n})$.

L'utilisation de la matrice M^T modifie les conditions suffisantes d'inclusion et d'exclusion des boîtes. Avec le système modifié, nous considérerons comme condition suffisante d'inclusion la formulation suivante :

$$\forall \mathbf{M} \in \mathbf{M}^{\mathbf{I}}, \forall \mathcal{F} \in \mathcal{F}^{\mathbf{I}}, \exists \mathbf{b} \in \mathbf{b}^{\mathbf{I}} : M\mathbf{b} = \mathcal{F} \quad \text{avec } \mathbf{b}^{\mathbf{I}} = \mathbf{D}^{\mathbf{I}}\tau_{\max} \quad (5.11)$$

Nous raisonnerons de la même manière pour les test d'exclusion.

5.6 Améliorer l'algorithme

Comprenons qu'améliorer un algorithme va dépendre du problème posé. Une "*légère*" modification de certains paramètres d'un problème peut conduire à des qualités de résultats et une vitesse de résolution très différentes. Aussi cette section consacrée à l'amélioration de l'algorithme consiste plus à une énumération d'astuces de programmation, et de méthodes de résolution. Le choix des méthodes utilisées définit alors une stratégie de résolution parmi la multitude possible.

5.6.1 Etape du point de choix

Nous pouvons choisir de calculer pour le centre de la boîte traitée les valeurs de τ . Si nous vérifions que le centre de la boîte n'est pas dans l'espace de travail statique alors, il n'est pas nécessaire d'effectuer un test d'inclusion puisque l'équation 5.2 ne sera jamais vérifiée. Dans le cas contraire, nous effectuerons le test d'inclusion. Nous rajoutons donc l'étape du point de choix juste après le test d'exclusion s'il n'est pas concluant. Suivant le problème posé, il peut-être plus intéressant d'effectuer ce test en priorité et suivant son résultat de se diriger vers un test d'inclusion ou d'exclusion.

5.6.2 Contraintes supplémentaires

Rajouter des contraintes supplémentaires peut également conduire à accélérer la résolution du système. Un parfait exemple va être la prise en compte des contraintes géométriques : $\rho_{min} \leq \rho_i \leq \rho_{max}$. Elles vont être testées en priorité et si elles amènent l'exclusion d'une boîte, elles évitent de réaliser un nombre important de tests d'exclusion, d'inclusion impliquant un grand nombre d'évaluations liées aux coefficients de la matrice. De manière générale, plus il y a de contraintes plus il y a des chances de voir des boîtes de tailles importantes être exclues.

5.6.3 Stockage de données

Nous avons vu qu'un algorithme de type *Branch and Prune* pouvait être vu sous la forme d'un arbre. Chaque feuille de l'arbre représentant une itération de l'algorithme sur une boîte B . Si cette boîte n'est ni exclue ni incluse alors elle sera bissectée et le même processus est réitéré. Nous appellerons les boîtes issues de B , les boîtes *filles* de B et B sera considérée comme la boîte *mère* des boîtes qu'elle a engendrée au cours du processus de résolution.

Si l'on suppose que certaines propriétés ou contraintes sont vérifiées pour une boîte B quelconque, alors pour toutes ses boîtes fils nous vérifierons les mêmes propriétés et les mêmes contraintes. Ainsi le stockage des propriétés, et des contraintes vérifiées évitera d'effectuer des calculs dont nous connaissons à l'avance le résultat, ce qui réduira le temps de calcul.

5.6.3.1 Stockage des valeurs des τ_i

Stocker les valeurs de τ^I obtenues pour la variable τ , par exemple après un test du type Gauss-Seidel, peut amener à améliorer le test par évaluation du torseur statique. Multiplier l'évaluation A^I de la transposée de la matrice jacobienne cinématique inverse par le vecteur intervalle τ^I conduit généralement à un meilleur résultat τ_{reduit}^I que celui obtenu après multiplication par τ_{max}^I (pour cela il suffit d'avoir réduit la largeur d'une composante de τ_{max}^I).

Bien évidemment il sert à améliorer l'estimé initial nécessaire à l'algorithme Gauss-Seidel. L'utilisation de τ_{reduit}^I à la place de τ_{max}^I conduira à une meilleure estimation à la fin d'une itération et permettra d'exclure ou d'inclure plus rapidement des boîtes.

5.6.3.2 Stockage du signe des dérivées

Nous avons vu que l'évaluation pouvait être améliorée si nous prenions en compte le signe des dérivées, voire des dérivées seconde. Il semble donc intéressant de conserver pour chaque boîte les résultats obtenus sur les boîtes précédentes. Si la dérivée d'une fonction est de signe constant pour une boîte, alors ce sera forcément le cas pour ses boîtes fils. il ne sera donc pas nécessaire de recalculer ces dérivées ce qui conduit à un gain significatif en temps de calcul.

5.6.3.3 Stockage des contraintes vérifiées

Intéressons-nous au cas où une boîte a vérifié certaines contraintes comme par exemple $\rho_{min} \leq \rho_1 \leq \rho_{max}$. Pour les boîtes fils, il n'est plus nécessaire de vérifier cette contrainte. Dans le cas présent, la première jambe du mécanisme ne conduira pas à une exclusion de la boîte. Nous pourrions donc supprimer un grand nombre de tests au cours du processus pour le traitement des boîtes suivantes suivant la nature du problème.

Nous pouvons également tirer parti du stockage des contraintes vérifiées lors de l'étape de bisection. En supposant que nous veuillons choisir comme variable à couper celle qui risque d'avoir le plus d'influence sur le résultat, nous regarderons l'influence des variables seulement sur les contraintes non vérifiées. Nous pouvons ainsi s'éviter un certain nombre de calculs suivant la stratégie choisie.

5.6.4 Améliorer les tests d'inclusions et d'exclusions impliquant la résolution du système linéaire

Il est possible d'améliorer ces tests suivant les cas étudiés si nous possédons certaines informations. Le meilleur moyen d'expliquer les améliorations proposées consiste à les illustrer par des exemples. Plaçons nous dans le cadre de la résolution d'un système linéaire de taille n .

Intéressons nous d'abord au cas du test d'exclusion. Le but de notre test est de vérifier que pour au moins l'une des composantes du vecteur intervalle des forces exercées par les jambes τ^I nous obtenons bien $|\tau_i^I| > \tau_{max}$. Il est important de ne pas évaluer tous les coefficients de la matrice jacobienne cinématique inverse, puis de résoudre le système linéaire et enfin de regarder s'il existe une composante i telle que $|\tau_i^I| > \tau_{max}$. Au lieu de cette méthode naïve, on effectue plutôt :

Etape 1 : Évaluation des coefficients de la première ligne de la transposée de la jacobienne cinématique sur la boîte traitée.

Etape 2 : Calcul de τ_1^I avec la méthode de Gauss-Seidel.

Etape 3 : test sur $|\tau_1^I|$

Si jamais $|\tau_1^I| > \tau_{max}$, nous pouvons affirmer que la boîte est *OUT*. Dans le cas le plus favorable, nous avons donc évité $(n - 1) \times n$ évaluations et $n - 1$ calculs de τ_1^I par rapport à la méthode naïve précédemment énoncée. Sinon continuons le même processus pour les autres lignes jusqu'à ce que l'une des composantes vérifie bien $|\tau_i^I| > \tau_{max}$.

Bien évidemment, nous ne nous trouvons pas forcément dans le cas le plus favorable. Cependant, au cours du processus de résolution, nous pouvons vérifier qu'une ou plusieurs composantes vérifient $|\tau_i^I| < \tau_{max}$. Elles ne vont donc pas conduire à l'exclusion de la boîte. Il peut alors être intéressant de ne pas les traiter en priorité et de considérer d'abord les composantes qui sont susceptibles de conduire à une exclusion de la boîte.

Si nous avons vérifié certaines contraintes du type $|\tau_i^I| < \tau_{max}$ et que $w(\tau_i^I)$ est *petit* alors nous pouvons considérer que la réduction de ces composantes ne va pas influencer "significativement" les résultats des autres composantes. Nous pouvons alors choisir de ne pas les traiter mais il faudra se montrer prudent avec ce genre de considération. En

effet, une petite réduction sur une composante $|\tau_i^I| < \tau_{max}$ pouvant parfois amener à une conclusion favorable du test d'exclusion pour les autres variables.

Comprenons qu'une méthode analogue pourra être utilisée avec le test d'inclusion. Nous traiterons alors en priorité les composantes qui ne vérifient pas $|\tau_i^I| < \tau_{max}$.

5.6.5 Ajout d'une méthode de type rognage

Il peut-être intéressant suivant le problème de s'aider d'une méthode de type rognage quand les boîtes deviennent *petite*. En effet nous aurons peu de chance d'exclure ou d'inclure une portion quand les boîtes vont être "grande", et cela risque d'amener à un surplus de temps de calcul pour notre cas dû aux nombreuses évaluations et tests impliqués. Nous allons détailler une adaptation de ce type de méthode dans le cadre restreint d'un résultat qui peut contenir boîtes intérieures (*IN*) et extérieures (*OUT*) en plus des boîtes frontières (*BORDER*).

5.6.5.1 Le rognage IN-OUT

La méthode de filtrage que nous allons décrire repose sur un principe fondamental de la résolution numérique de systèmes en analyse par intervalle que nous trouvons dans le travail de Hladik, Daney et Tsigaridas [21].

Principe fondamental : Une boîte *IN* et une boîte *OUT* ne peuvent pas être en contact. Il existe forcément au moins une boîte *BORDER* qui empêche tout contact entre les deux boîtes *IN* et *OUT*.

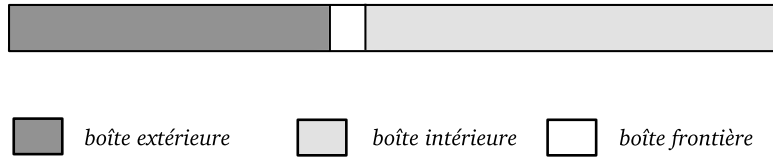


FIG. 5.3: Principe fondamental : une boîte *BORDER* empêche tout contact entre une boîte *IN* et une boîte *OUT*

En utilisant ce principe fondamental, nous avons observé certaines propriétés qui peuvent se révéler intéressantes car permettent d'éviter certaines étapes d'un processus de rognage classique.

5.6.5.2 Propriétés dérivant du principe fondamental

Avant d'énoncer ces propriétés, rappelons quels peuvent être les différents résultats obtenus lors d'une méthode de filtrage de type "shaving" classique. Pour notre méthode, il est important de détailler un peu plus précisément le cas où nous réduisons partiellement la boîte. Aussi, nous le séparons en deux sous-cas, à savoir celui où la boîte est réduite seulement pour une variable et le cas où la boîte est réduite pour au moins deux variables :

- *Cas 0* : Nous avons complètement éliminé la boîte initiale $\mathbf{x}_{init}^I = (X_1, \dots, X_n)$ de

dimension n . Soit cette dernière est *IN* (*Cas 0* : succès total du test d'inclusion), soit elle est *OUT* (*Cas 0 bis* : succès total du test d'exclusion).

- *Cas 1* : Nous n'avons pas pu réduire la boîte $\mathbf{x}_{\text{init}}^I$.

- *Cas 2* : Réduction partielle de $\mathbf{x}_{\text{init}}^I$ mais seulement pour une variable.

- a. Un seul type de test a fonctionné, nous avons inclus une ou plusieurs portions de la boîte (*Cas 2.a*) ou alors il a été exclu une ou plusieurs portions de la boîte (*Cas 2.abis*).

- b. A la fois le test d'inclusion et d'exclusion ont fonctionné pour une variable. Nous avons donc inclus une portion d'un coté et exclu une portion de l'autre.

- *Cas 3* : Réduction partielle de $\mathbf{x}_{\text{init}}^I$ pour au moins deux variables. Nous allons montrer qu'il est impossible d'avoir réussi à simultanément inclure et exclure une portion dans ce cas. Nous distinguerons donc les cas d'inclusion (*Cas 3*) et d'exclusion (*Cas 3 bis*).

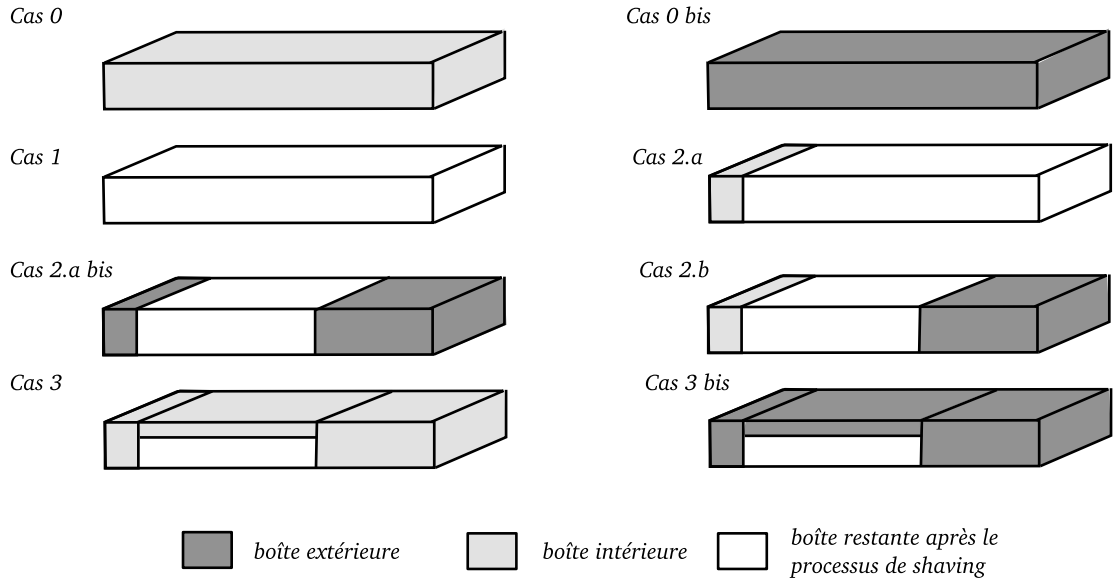


FIG. 5.4: Les différents cas possibles obtenus après une méthode de rognage

Propriété : Si au cours d'un procédé de shaving, une boîte a été réduite simultanément pour deux variables différentes (*cas 3*) alors forcément nous nous trouvons dans un cas où un seul type de test a fonctionné. En résumé, les portions éliminées de la boîte, sont soit intérieures, soit extérieures.

Démonstration : Soit $\mathbf{x}_{\text{init}}^I = (X_1, \dots, X_n)$ la boîte initiale de dimension n avant le processus de rognage, et soit la boîte $\mathbf{x}_{\text{new}}^I = (X_1, \dots, X'_i, \dots, X'_j, \dots, X_n)$ celle obtenue après réduction sur les deux variables i et j .

Sans alourdir le propos par un excès de généralité, supposons que l'on a pu inclure pour la variable i , une tranche T_i correspondant à la portion $[x_i, x_i + \epsilon]$ de la boîte initiale. Nous avons choisi ici la portion inférieure associée à la i -ème composante mais nous pourrions tout aussi bien traiter la portion supérieure. Si nous avons éliminé cette

tranche en traitant la variable i alors la tranche T_j éliminée avec la variable $j \neq i$ est forcément en contact avec T_i . D'après le principe fondamental, T_i et T_j sont donc deux boîtes *IN*. On raisonne de la même manière si on a supposé T_i extérieure alors T_j est *OUT*. \square

Conséquences :

- Intéressons au cas 2b. et considérons le cas où pour une variable i donnée le test d'inclusion a été concluant à gauche ($\mathbf{x}_{I_i}^I$ inclus) et le test d'exclusion concluant à droite ($\mathbf{x}_{I_i}^I$ est une boîte extérieure). En suivant le raisonnement précédent, nous comprenons qu'il est impossible de réduire à nouveau la boîte en appliquant le rognage sur les variables $i + 1, \dots, n$ car les boîtes alors impliquées dans le rognage seraient forcément en contact avec une boîte intérieure et une boîte extérieure. Nous pouvons donc arrêter le processus de rognage dès que nous rencontrons le cas 2b. Autre aspect intéressant de ce cas : la bisection de la boîte restante devra se faire sur la variable i , (sous réserve que $w(X_i) > \epsilon$) car une bisection sur une autre variable conduirait à des boîtes impossible à inclure ou exclure d'après le principe fondamental. Si $w(X_i) < \epsilon$ alors il n'est plus nécessaire de bissecter les autres variables, nous pouvons déjà stocker la boîte comme une boîte *BORDER*.

- Dès qu'on exclut des portions de la boîte initiale sur deux variables différentes, alors dans le cadre du processus de rognage pour toutes les variables il suffit de se borner au seul test d'exclusion.

- Dès qu'on inclut des portions de la boîte initiale sur deux variables différentes, alors dans le cadre du processus de rognage pour toutes les variables il suffit de se borner au seul test d'inclusion.

- Au cours du processus de rognage si on a exclu une ou des portions de la boîte initiale sur une seule variable alors pour les autres variables il suffit de se borner au seul test d'exclusion.

- Au cours du processus de rognage si on a inclus une ou des portions de la boîte initiale sur une seule variable alors pour les autres variables il suffit de se borner au seul test d'inclusion.

5.6.5.3 Implantation d'un algorithme de rognage *IN-OUT*

Sur le fond, il est identique à un algorithme de rognage classique. Nous y ajoutons le stockage des variables pour lesquelles le rognage s'est montré efficace et la prise en compte des propriétés précédemment énoncées. Concernant le stockage, précisons également si la portion ou les portions de boîtes éliminées ont été exclues ou incluses. Nous avons choisi d'associer un tableau d'entier de dimension n à chaque boîte \mathbf{x}^I de dimension n . Ce tableau nous permet de retrouver quelles sont les éléments de la boîtes en contact avec une boîte intérieure ou extérieure. Nous pouvons donc savoir quel test est à utiliser lors du processus de rognage : celui d'exclusion ou celui d'inclusion voire dans le cas 2.b abandonner le rognage.

Pour la i -ème variable, si nous ne sommes pas parvenu à exclure une portion, la i -ème valeur du tableau sera 0. Si nous avons exclu la portion de gauche, la i -ème valeur sera

−1. Si nous avons exclu la portion de droite, la i -ème valeur sera −2 et si nous avons exclu la portion de droite et la portion de gauche nous fixerons la i -ème valeur à −3. De la même manière pour le cas d'inclusion on utilisera les nombres entiers positifs 1, 2 et 3 pour remplir le tableau. Enfin pour le cas très particulier (cas2.b) où nous avons obtenu pour une variable du *IN* à gauche et du *OUT* à droite fixons la i -ème valeur à 99, et si nous avons du *OUT* à gauche et du *IN* à droite la i -ème valeur sera −99.

Lors de l'étape de bisection, il faudra faire attention à bien mettre à jour le tableau. Nous pourrions également mettre à jour le tableau si nous parvenons à contracter (avec le test d'exclusion de Rohn par exemple) une boîte au cours du processus de résolution. Nous pourrions à l'aide du tableau s'éviter l'étape du point de choix car le tableau nous précise immédiatement si nous avons une chance d'inclure la boîte entièrement ou non (il existe une variable pour laquelle la boîte est en contact avec une boîte *OUT*) ou de réduire au moins partiellement la boîte (il existe au moins 2 variables pour lesquelles la boîte est en contact avec une ou des boîtes *IN*). Dans le cas favorable, nous pourrions donc s'éviter soit un test d'exclusion soit un test d'inclusion.

Nous voyons que l'algorithme de rognage *IN-OUT* a également des répercussions sur l'algorithme général car le stockage des boîtes en contact permet d'éviter dans les cas favorables certains tests d'inclusion ou d'exclusion. De plus connaître quel type de boîte est en contact avec la boîte traitée peut conduire (suivant le problème) à répondre à la question déjà abordée : lequel des intervalles X_{l1} ou X_{r1} doit-il être traité en priorité ? Rappelons que X_{l1} et X_{r1} sont les fils de l'intervalle X_1 après sa bisection.

5.6.6 Bisection multi-niveaux

Dans l'algorithme précédent il peut arriver que les intervalles pour \mathcal{F}^I soient si larges que même pour une pose χ nous ne puissions pas décider si χ est inclus dans W_S . La décision sur la boîte ne pourra donc être prise que par une boucle interne de "*branch and bound*" qui portera sur \mathcal{F}^I . Si au cours de cette boucle, nous trouvons que la pose est à l'intérieur de W_S pour tous les éléments B_F de \mathcal{F}^I alors nous concluons que la pose χ est intérieure à W_S . Par contre, si pour une boîte B_F nous trouvons qu'il existe $\tau_i > \tau_{max}$ alors nous aurons prouvé que la pose χ est extérieure à l'espace de travail statique.

Plus généralement, nous ne cherchons pas à exclure une seule pose mais bien des ensembles de poses, nous allons donc définir un algorithme multi-niveaux de bisection. Il sera constitué de deux passes de type "*branch and bound*", l'une interne concerne les \mathcal{F}^I et l'autre externe concerne les boîtes \mathbf{X}^I associées aux variables χ .

5.6.6.1 La structure pour le multi-niveaux

Définissons d'abord la structure que nous allons utiliser pour implanter notre algorithme. On dispose d'une liste L_X de boîtes \mathbf{X}^I associées aux variables correspondant à χ . \mathbf{X}_i^I désigne le i -ème élément de L_X . A chacune des boîtes \mathbf{X}_i^I nous associons une liste $L_F(\mathbf{X}_i^I)$ de boîtes sur \mathcal{F} qui seront utilisées par la boucle interne pour vérifier la validité des τ_i . Désignons par F_{ij}^I le j -ème élément de la liste $L_F(\mathbf{X}_i^I)$. Cette structure est illustrée de manière graphique avec la figure 5.5.

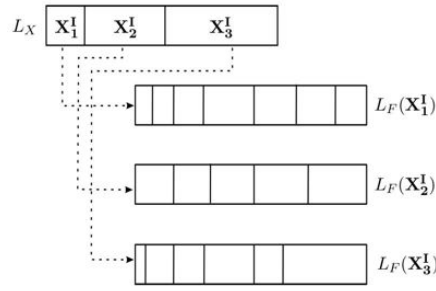


FIG. 5.5: La structure de données pour notre algorithme *Branch and Prune* multi-niveaux

5.6.6.2 Algorithme multi-niveaux

La structure décrite dans la section précédente permet de proposer un algorithme multi-niveaux pour le calcul de l'espace de travail statique. Nous définissons ici deux niveaux, l'un pour les boîtes X^I et l'autre pour les boîtes \mathcal{F}^I . Désignons par ϵ_x le diamètre maximal pour une boîte X^I et par ϵ_F le diamètre maximal pour une boîte \mathcal{F}^I et n_F correspond au nombre de boîtes dans la liste $L_F(X_i^I)$.

Premier niveau : parcours de la liste L_x .

Etape 1 : calculons un nombre d'itération maximum $iter_{max}$. La manière dont nous calculerons le nombre maximum d'itération favorisera un niveau plutôt qu'un autre. Plus le nombre $iter_{max}$ est grand et plus nous favoriserons la bisection sur les boîtes associées à \mathcal{F}^I .

Etape 2 : lançons une procédure associée au second niveau. Nous effectuerons au maximum $iter_{max}$ itérations de ce processus. Nous cherchons à exclure ou inclure X_i^I .

Second niveau : parcours de $L_F(X_i)$ qui contient n_F éléments

Il débute avec un index $j = 1$ et un compteur de boucle $iter = 1$.

Etape A : Si $iter > iter_{max}$ alors nous passons à l'étape 3. Sinon on passe à l'étape B.

Etape B : effectuons un test d'exclusion avec X_i^I, F_{ij}^I

Si le test d'exclusion est concluant, stockons X_i^I dans OUT supprimons la de L_x . Passons à la boîte suivante de L_x , retour à l'étape 1.

Si le test d'exclusion n'est pas concluant, passons à l'étape C.

Etape C : effectuons un test d'inclusion avec X_i^I et F_{ij}^I Si le test est concluant, on distingue deux cas : $n_F = 1$ et $n_F > 1$

- $n_F = 1$: stockons X_i^I dans IN et supprimons la de L_x . Retour à l'étape 1.

- $n_F > 1$: retour à l'étape A après avoir supprimé F_{ij}^I de $L_F(X_i)$.

Mise à jour de $n_F = n_F - 1$ et $iter = iter + 1$.

Sinon passage à l'étape suivante.

Etape D : Si $Diam(F_{ij}) > \epsilon_F$: bisection de F_{ij}^I

Mise à jour de $n_F = n_F + 1$, $iter = iter + 1$ et retour à l'étape A

Sinon distinguons plusieurs cas : $j < n_f$ et $j = n_f$

- Si $j < n_f$ alors $j = j + 1$
- Si $j = n_f$, passage à l'étape 3.

Etape 3 : Si $Diam(X_i) < \epsilon_X$ stockons X_i^I et $L_F(X_i)$ dans *BORDER* et supprimons ces éléments. Retour à l'étape 1. Sinon bisection de X_i^I et retour à l'étape 1.

Le processus s'arrête quand il n'y a plus de boîte dans la liste L_x . Nous venons de donner le squelette de l'algorithme multi-niveaux et nous pourrions définir un grand nombre de stratégies avec cet algorithme de base.

5.7 Adaptation de l'algorithme pour répondre à différents problèmes

5.7.1 Prise en compte d'incertitude sur les paramètres géométriques

Notre but est d'obtenir l'espace de travail statique tout en prenant en compte les incertitudes liées, par exemple, aux tolérances de fabrication du mécanisme. Au lieu de considérer un robot parfait avec des points d'attaches A_i et B_i dont les coordonnées sont supposées parfaitement connues, nous remplaçons les coordonnées de ces points par des intervalles. Si \mathbf{P}^I désigne l'ensemble de ces paramètres géométriques, alors calculer l'espace statique en prenant en compte les incertitudes sur les paramètres géométriques revient à déterminer l'ensemble :

$$W_s = \{\chi \in W : \forall \mathbf{P} \in \mathbf{P}^I, \forall \mathcal{F} \in \mathcal{F}^I \|\tau\|_\infty \leq \tau_{max}, \text{ et vérifie } \mathbf{J}^{-T}(\chi, \mathbf{P})\tau = \mathcal{F}\} \quad (5.12)$$

Le calcul de l'espace de travail statique avec la prise en compte des incertitudes sur les paramètres géométriques, ne nécessite pas de modifications majeures de notre algorithme. Il suffit de prendre en compte les incertitudes sur les coordonnées des points d'attache lors de l'étape d'évaluation de la transposée de la matrice jacobienne cinématique. Les tests d'inclusion et d'exclusion restent identiques.

Il est également possible de définir un algorithme multi-niveaux si l'on souhaite également effectuer des bisections sur l'ensemble des paramètres géométriques (par exemple dans le cas où les incertitudes sont "*grandes*") voire sur les variables qui correspondent à la charge appliquée à la plate-forme mobile. Nous utiliserons donc une structure de ce type :

- Une liste L_X qui correspond à l'ensemble des poses χ que nous souhaitons étudier.
- Pour chaque élément X_i de L_x , une liste $L_{P,F}(X_i)$ qui contient l'ensemble des paramètres géométriques possibles et des charges appliquées à l'organe terminal.

Nous pourrions alors définir un algorithme sur 2 niveaux, le premier associé aux variables χ , le second aux paramètres géométriques et à la charge appliquée sur la plate-forme. De la même manière que pour l'algorithme multi-niveaux précédemment décrit il suffira pour exclure une boîte \mathbf{x}^I , qu'un test d'exclusion impliquant un élément de $L_P(X_i)$ et un élément de $L_{P,F}(X_i)$ soit concluant. Par contre pour inclure une boîte, il faudra que tous les éléments de $L_{P,F}(X_i)$ vérifient les conditions d'inclusions.

5.7.2 Vérification de l'inclusion d'une zone dans W_s

Il peut être également intéressant de vérifier qu'une zone donnée est incluse dans l'espace de travail afin de vérifier qu'une trajectoire passe assez "loin" des singularités ou pour comparer plusieurs manipulateurs différents en vue du design d'un manipulateur. Dans ce cas dès que nous trouvons une boîte frontière ou une boîte extérieure, nous n'avons plus besoin de traiter les autres boîtes restantes dans la liste L puisque nous y avons déjà répondu à la question posée. Notre objectif est d'obtenir une réponse négative le plus rapidement possible, sachant que si le résultat est positif il faudra de toute manière vérifier l'ensemble du domaine. Nous chercherons donc à atteindre le plus rapidement possible une boîte frontière ou une boîte extérieure. Nous voyons ici qu'un parcours en largeur pourra potentiellement conduire à répondre plus rapidement à la question posée. Enfin nous pourrions nous aider de méthode comme le rognage $IN - OUT$ afin de privilégier une boîte plutôt qu'une autre après le procédé de bisection.

5.8 Résultats

Les résultats qui concerne le robot 3 – RPR sont illustrés pour le même mécanisme que dans le chapitre 3, on aura donc comme paramètre géométrique $OA_1 = (0, 0)$, $OA_2 = (20, 0)$, $OA_3 = (12, 10)$ pour les points d'attache à la base et $CB_{r1} = (-4, 4)$, $CB_{r2} = (4, -4)$, $CB_{r3} = (0, 2)$ pour les points d'attache à la plate-forme mobile. Les résultats qui concerne la plate-forme de Gough utilisent les paramètres géométriques suivant : $OA_1 = (300, 0, 0)$, $OA_2 = (150, 250, 0)$, $OA_3 = (-150, 250, 0)$, $OA_4 = (-300, 0, 0)$, $OA_5 = (-150, -250, 0)$, $OA_6 = (150, -250, 0)$ et $CB_{r1} = (200, 0, 0)$, $CB_{r2} = (100, 150, 0)$, $CB_{r3} = (-100, 150, 0)$, $CB_{r4} = (-200, 0, 0)$, $CB_{r5} = (-100, -150, 0)$, $CB_{r6} = (100, -150, 0)$

Le tableau 5.1 nous donne les différents paramètres impliqués dans le calcul de l'espace de travail statique du robot 3 – RPR . P_{geo} désigne les incertitudes sur les paramètres géométriques liées aux tolérances de fabrication. La taille minimum des boîtes \mathbf{x}^I associée aux poses du mécanisme est donnée par ϵ_x . Le tableau 5.3 nous donne les différents paramètres impliqués dans le calcul de l'espace de travail statique de la plate-forme de Gough avec une valeur du $\tau_{max} = 10$.

figure	\mathcal{F}^I	τ_{max}	θ (rad)	\mathbf{x}_{init}^I	ϵ_x	P_{geo}
5.6	$(0 \pm 0.1, 1.5 \pm 1.5, 0 \pm 0.1)$	8	-0.1	$([-16, 20], [-25, 20])$	0.5	Non
5.7	$(0 \pm 0.1, 1.5 \pm 1.5, 0 \pm 0.1)$	8	-0.1	$([-16, 20], [-25, 20])$	0.1	Non
5.8	$(0 \pm 0.1, 1.5 \pm 1.5, 0 \pm 0.1)$	8	-0.1	$([-16, 20], [-25, 20])$	0.01	Non
5.9	$(1 \pm 1, 1 \pm 1, 1 \pm 1)$	3	0.1	$([-10, 25], [-18, 23])$	0.1	± 0.1

TAB. 5.1: Paramètres utilisés pour tracer les différentes courbes pour le robot 3 – RPR

Comme il n'est pas évident d'observer de manière graphique le résultat en 3 dimensions sur du papier, nous proposons des coupes planes mais il faut bien comprendre que l'espace de travail statique peut s'obtenir en n dimension dans des fichiers.

Les figures 5.6, 5.7 et 5.8 représentent des sections planes obtenues pour des valeurs différentes du seuil ϵ_x . Les boîtes qui sont grisées correspondent aux boîtes intérieures

figure	\mathcal{F}^I	θ_{Euler} (rad)	\mathbf{x}_{init}^I	ϵ_x
5.10	$\forall i \neq 3 F_i = 0 \pm 0.1, F_3 = [0.3]$	$(\pi/6, 0, 0)$	$(0 \pm 600, 0 \pm 600, 400)$	5
5.11	$\forall i \neq 3 F_i = 0 \pm 0.1, F_3 = [0.3]$	$(\pi/6, 0, 0)$	$(0 \pm 600, 0 \pm 600, 400)$	1
5.12	$\forall i \neq 3 F_i = 0 \pm 0.1, F_3 = [0.3]$	$(\pi/6, 0, 0)$	$(0 \pm 600, 0 \pm 600, 400)$	0.5

TAB. 5.2: Paramètres utilisés pour tracer les différentes courbes pour la plate-forme de Gough avec $\tau_{max} = 10$

figure	V_{IN}	V_{BORDER}	$I_{quality}$	temps de calcul total (s)
5.6	360	397	0.47	< 1
5.7	389	200	0.66	6
5.8	420	160	0.72	30
5.10	264507	123772	0.68	2
5.11	300593	61190	0.83	63
5.12	303029	55234	0.84	182

TAB. 5.3: Indication sur la qualité et temps de calcul total

et celles qui sont laissées blanches sont les extérieures. Les espaces blancs que nous observons sur les trois figures correspondent aux boîtes frontières que nous avons choisi de ne pas représenter afin de ne pas surcharger la lecture des résultats. Nous observons que l'approximation de l'espace de travail statique est composé pour cet exemple précis de trois composantes disjointes pour les trois figures. Plus nous diminuons le seuil ϵ_x et plus l'approximation de W_s se raffine. On remarque qu'entre la figure 5.6 et 5.8 de nombreuses boîtes extérieures sont apparues, ce qui améliore la qualité de notre approximation. Précisons que les temps de calcul sont différents suivant la valeur de ϵ_x , nous obtenons le résultat de 5.6 en moins d'une seconde alors qu'il nous faut 30 secondes pour obtenir le résultat correspondant à 5.8.

La figure 5.9 est intéressante car elle montre l'influence des incertitudes sur les paramètres géométriques. En blanc et gris clair nous pouvons observer les boîtes *OUT* et *IN* obtenues pour un robot "parfait" tandis qu'en gris foncé nous avons représenté les boîtes intérieures obtenues pour le même robot mais avec la prise en compte des incertitudes sur les paramètres géométriques. Nous voyons que l'approximation de l'espace de travail statique obtenu pour un robot non parfait est bien contenue dans l'approximation de W_s pour le robot parfait. Nous obtenons ces résultats respectivement en 10 secondes (robot parfait) et 30 secondes (avec les incertitudes).

Nous pouvons observer les mêmes effets avec les différents exemples (figures 5.10, 5.11 et 5.12) qui représentent l'approximation de W_s pour la plate-forme de Gough. Nous voyons donc que notre algorithme est efficace pour tout type de robot parallèle. Remarquons que les temps de calculs sont plus importants que pour le 3 – *RPR* (voir le tableau 5.3), nous comprendrons que cela est dû au nombre plus important de coefficients de la matrice jacobienne que nous devons évaluer et à la taille du système linéaire que nous résolvons. Précisons que nous avons également obtenu des résultats pour une boîte initiale de dimension > 2 . Nous avons choisi d'afficher le temps de calcul total sur le tableau 5.3, pour obtenir les temps intermédiaires qui sont obtenus avec les boîtes *BORDER* précédemment obtenus, il suffit de retrancher le temps de calcul total du

résultat précédent au temps de calcul total. Précisons que nous avons choisi de ne pas illustrer des résultats en dimension 3 ou plus, mais avec les mêmes paramètres que pour la figure 5.10 excepté $\mathbf{x}_{\text{init}}^{\text{I}} = [-600, 600] \times [-600, 600] \times [300, 500]$ nous obtenons par exemple en 190 secondes un résultat avec une qualité d'approximation $I_{\text{quality}} = 0.61$ ($V_{\text{IN}} = 4400000$ et $V_{\text{BORDER}} = 2200000$).

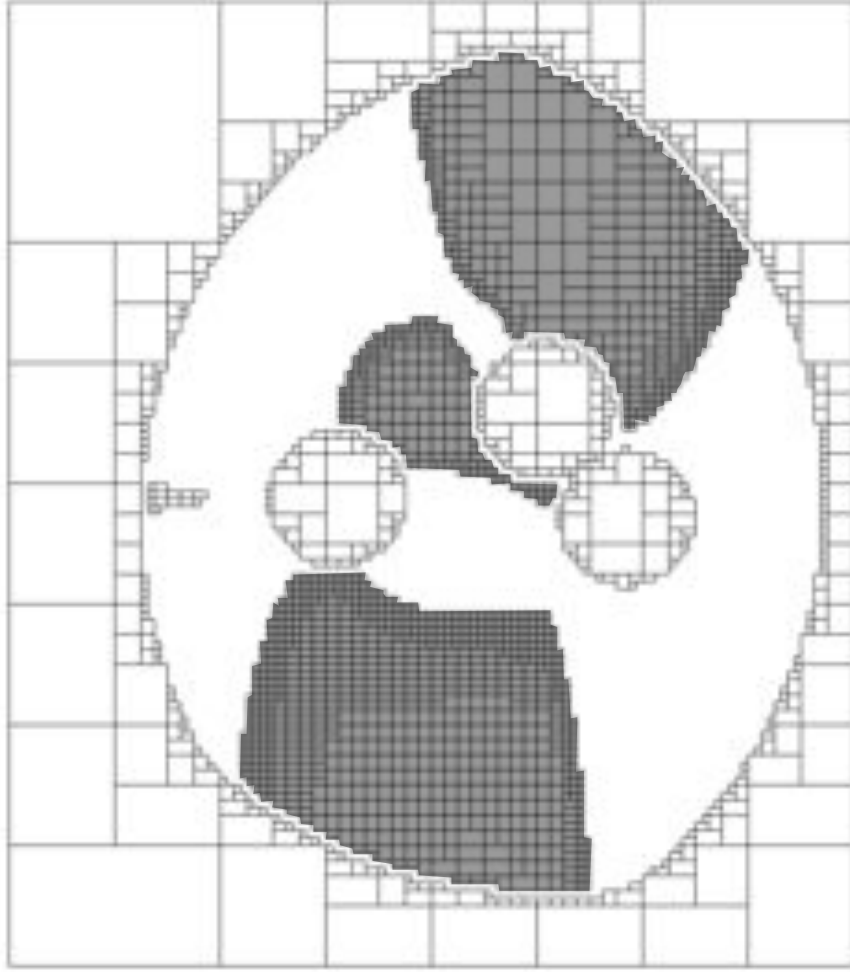


FIG. 5.6: *Approximation de l'espace de travail statique. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. Approximation pour $\mathbf{x}_{\text{init}}^{\text{I}} = [-16, 20] \times [-25, 20]$ et pour des tailles de boîtes minimum $\epsilon_x = 0.5$. Pour le reste des paramètres se référer au tableau 5.1.*

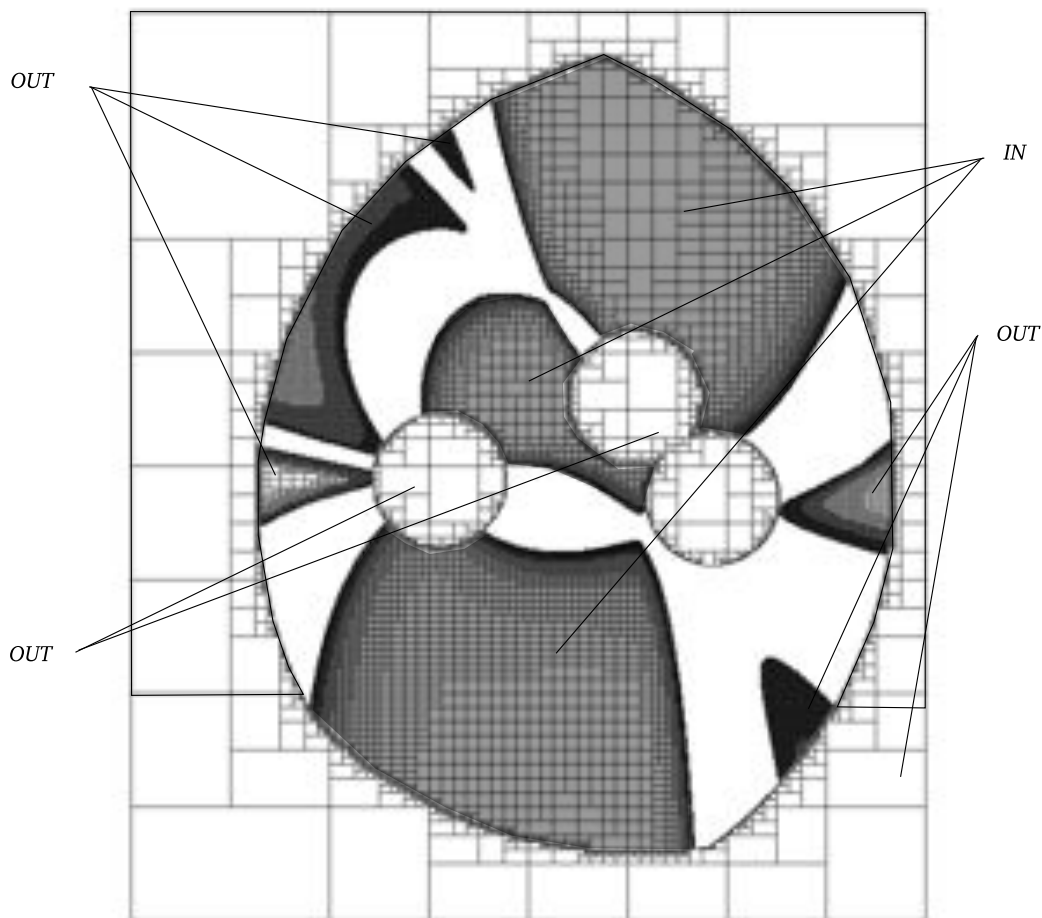


FIG. 5.7: *Approximation de l'espace de travail statique. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. Approximation pour $\mathbf{x}_{\text{init}}^{\text{I}} = [-16, 20] \times [-25, 20]$ et pour des tailles de boîtes minimum $\epsilon_x = 0.1$. Pour le reste des paramètres se référer au tableau 5.1.*

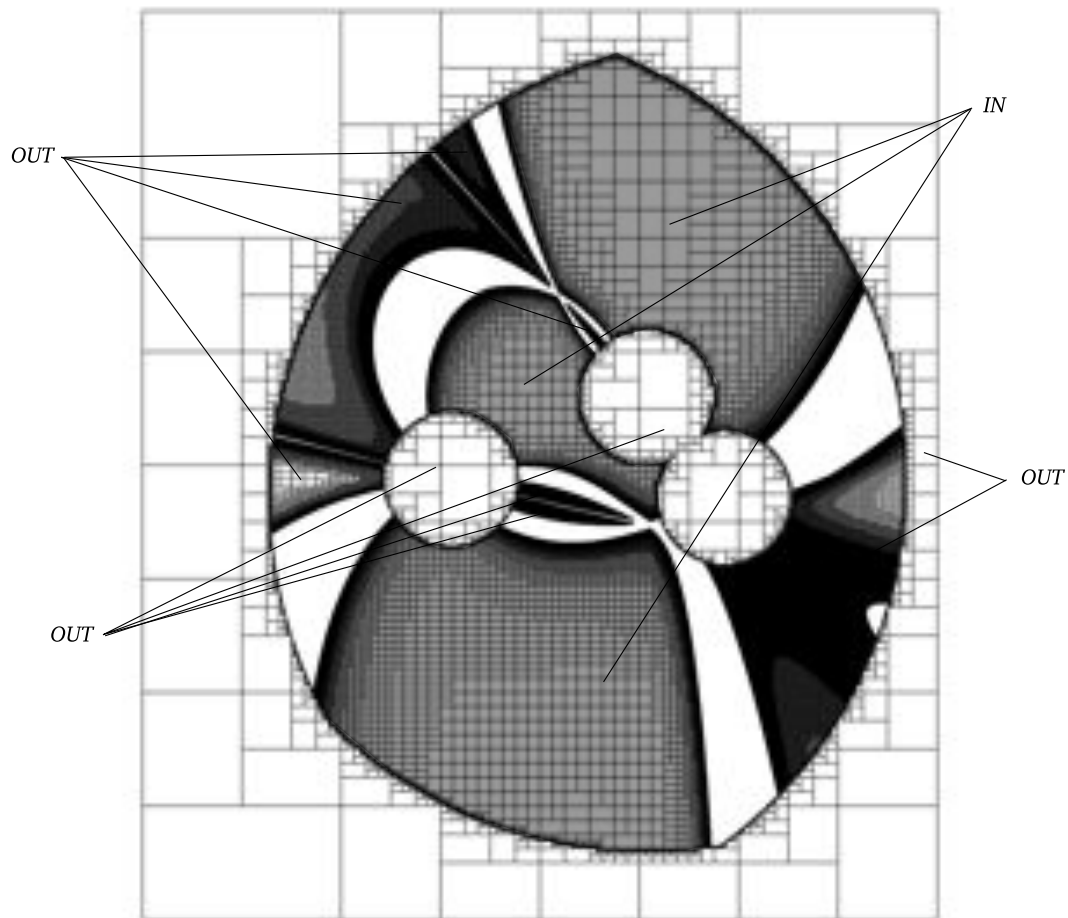


FIG. 5.8: Approximation de l'espace de travail statique. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. Approximation pour $\mathbf{x}_{\text{init}}^{\text{I}} = [-16, 20] \times [-25, 20]$ et pour des tailles de boîtes minimum $\epsilon_x = 0.01$. Pour le reste des paramètres se référer au tableau 5.1.

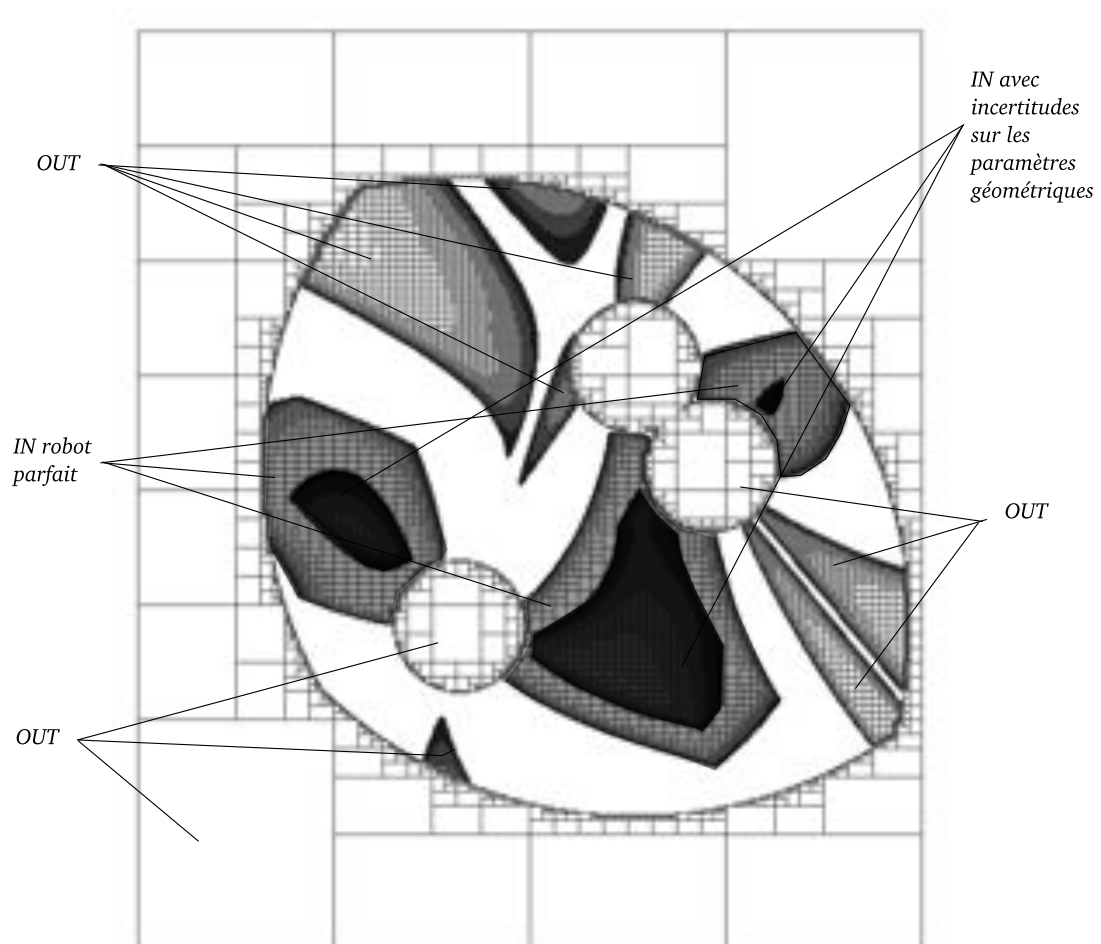


FIG. 5.9: Comparaison entre l'espace de travail statique pour un robot parfait et un robot réel. En blanc les boîtes OUT, en gris clair les boîtes IN pour le robot parfait et en gris foncé les boîtes IN si l'on tient compte des incertitudes sur les paramètres géométriques. L'espace laissé vide correspond aux boîtes BORDER. Pour le reste des paramètres se référer au tableau 5.1.

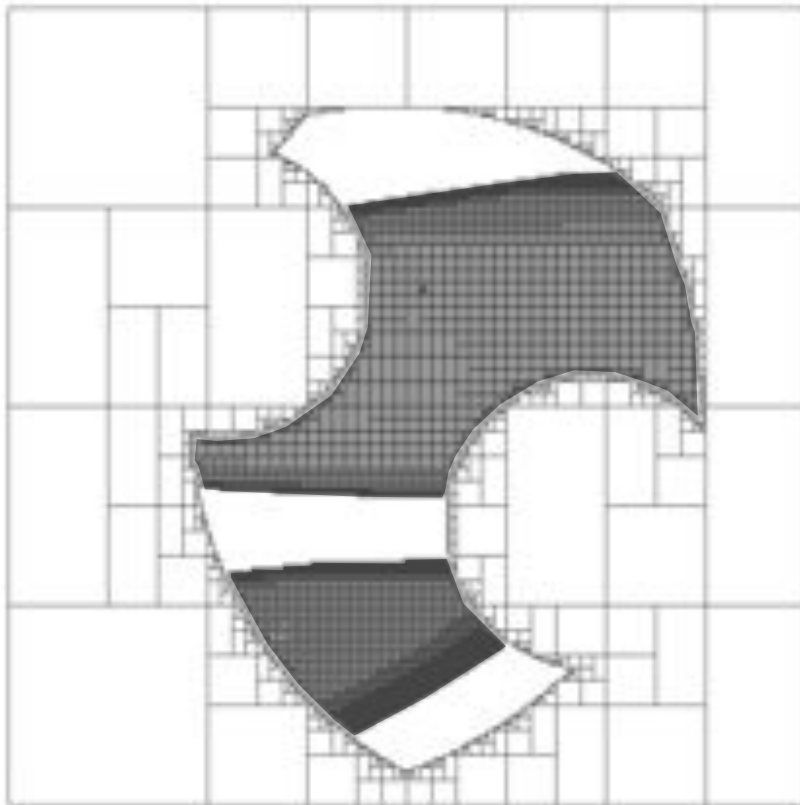


FIG. 5.10: Coupe plane $x \times y = [-600, 600] \times [-600, 600]$ de W_s pour la plate-forme de Gough. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. la taille minimum des boîtes est $\epsilon_x = 5$.

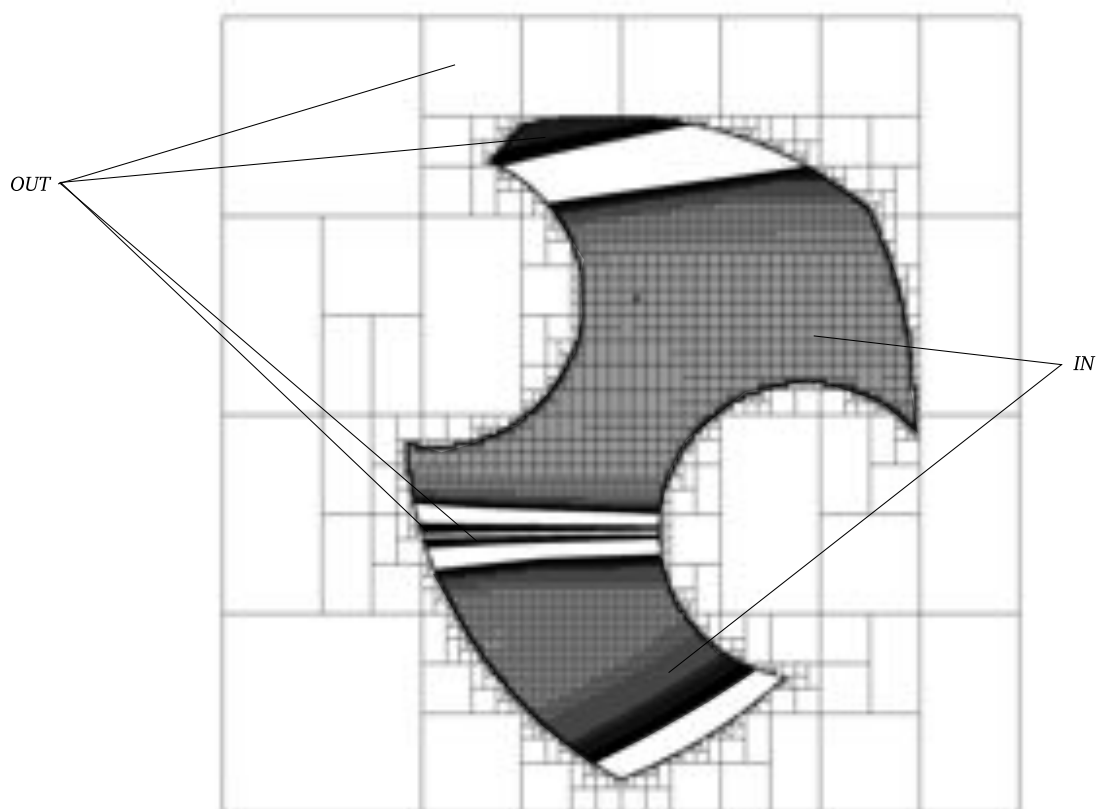


FIG. 5.11: Coupe plane $x \times y = [-600, 600] \times [-600, 600]$ de W_s pour la plate-forme de Gough. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. la taille minimum des boîtes est $\epsilon_x = 1$.

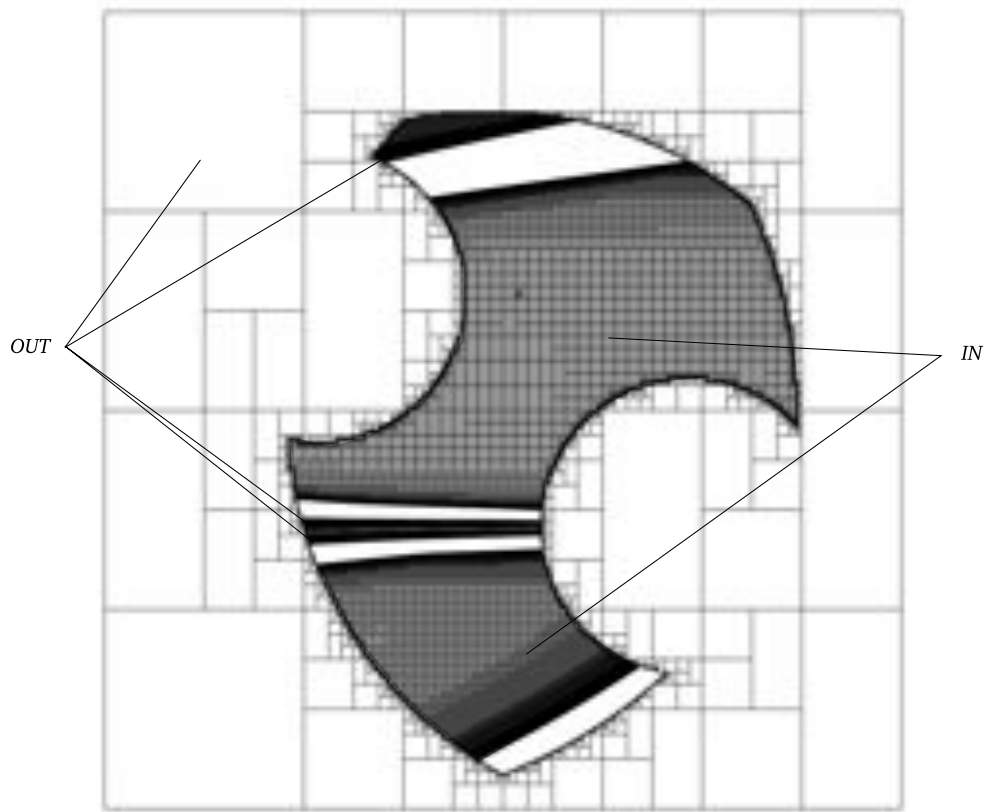


FIG. 5.12: Coupe plane $x \times y = [-600, 600] \times [-600, 600]$ de W_s pour la plate-forme de Gough. Les boîtes grises sont IN et les blanches OUT. L'espace laissé vide correspond aux boîtes BORDER. la taille minimum des boîtes est $\epsilon_x = 0.5$

5.9 Conclusion

Dans ce chapitre, nous venons de proposer un algorithme qui nous permet de calculer l'espace de travail statique dans sa globalité et non plus des sections planes. De plus, nous avons calculé W_s non plus pour une charge donnée mais un ensemble de charges et notre algorithme permet d'obtenir les résultats pour des robots parallèles sans articulations passives.

Nous avons proposé un moyen de mesurer la qualité de notre approximation et nous avons donné un grand nombre de méthodes, d'améliorations qui permettent de définir une multitude de stratégies de résolution. Parmi ces méthodes, nous avons proposé une adaptation d'une méthode de type rognage pour les cas d'un continuum de solution.

Nous avons également montré que sans modifications majeurs de notre algorithme, nous pouvions prendre en compte les incertitudes liées à la tolérance de fabrication du mécanisme afin de calculer une approximation certifiée de W_S .

Enfin nous sommes également capable de vérifier qu'une zone donnée est bien incluse dans W_s et cela peut s'avérer utile pour de la synthèse de mécanisme mais également pour vérifier si une trajectoire est sûre d'un point de vue des singularités.

CONCLUSION ET PERSPECTIVES

6

6.1 Conclusion

Ce document a abordé le problème des singularités pour les robots parallèles. Nous avons rappelé dans le second chapitre quelles sont les méthodes utilisées pour étudier ces configurations particulières. Ces méthodes permettent notamment de répondre à la question : "le robot se trouve-t-il en position singulière ?" Nous avons montré que répondre à cette question n'était pas suffisant car des poses du mécanisme non singulières peuvent présenter des problèmes de contrôlabilité ou être soumises à des contraintes internes élevées, dès lors qu'elles sont "proches" d'une singularité. Il convient alors de s'interroger sur la notion de "*distance*" aux singularités. Nous avons abordé cette question du point de vue de la statique et avons observé qu'à proximité d'une pose singulière l'intégrité physique d'un robot parallèle peut être menacée. Nous avons alors considéré la zone d'influence statique autour d'une ou plusieurs singularités comme l'ensemble des poses du mécanismes pour lesquelles le robot parallèle risque de subir des détériorations.

Afin de caractériser la zone d'influence statique autour de ces configurations particulières, nous avons défini un indice de proximité aux singularités dans le chapitre 3. Notre indice repose sur un critère physique simple : "éviter toute détérioration du robot". Il faut donc que les efforts dans les jambes du mécanisme soient inférieurs, en valeur absolue, à un seuil donné. Ce seuil correspond à la force ou au couple de rupture du plus "faible" composant mécanique d'une jambe.

Avec cette indice nous avons pu caractériser la zone d'influence statique des singularités du manipulateur et ainsi définir un espace où l'intégrité du manipulateur est préservée. *L'espace de travail statique* W_S est la zone de l'espace de travail où le manipulateur peut évoluer sans craindre la moindre détérioration, c'est-à-dire l'ensemble des poses pour lesquelles les efforts dans les jambes sont tous inférieurs en valeur absolue au seuil de rupture du plus faible des composants des jambes. Nous avons proposé deux algorithmes pour calculer W_S ainsi que des variantes permettant de vérifier qu'une trajectoire ou un espace donné est à l'intérieur de l'espace de travail statique, ou qui permettent de prendre en compte différentes contraintes sur le fonctionnement du robot.

Il est apparu clairement sur les résultats que cette manière pratique et réaliste d'aborder les singularités a une influence considérable sur l'espace de travail effectif du robot. Si on considère un mécanisme dont l'espace de travail est traversé par une hypersurface singulière sans pour autant le séparer en plusieurs composantes, il est possible que l'espace de travail statique soit constitué par différentes composantes disjointes. En effet W_S est généralement de taille bien plus faible que l'espace de travail auquel on a soustrait l'hy-

persurface singulière. Aussi la notion d'espace de travail statique peut avoir un impact important sur l'utilisation du robot.

6.2 Perspectives

En ce qui concerne les perspectives de cette étude, nous pouvons dégager un certain nombre de points qu'il nous paraîtrait intéressant d'étudier :

- définir un indice de proximité aux singularités d'un point de vue cinématique. Il s'agirait ici de fixer une limite sur les vitesses articulaires et de définir un *espace de travail cinématique* comme l'ensemble des poses où pour un ensemble de torseurs cinématiques donné, les vitesses articulaires, en valeur absolu, n'excède pas un seuil donné. La relation vitesse articulaire, torseur cinématique étant

$$\dot{\mathbf{Q}} = J^{-1}\mathbf{v} \quad (6.1)$$

On se rend compte que nos algorithmes pourraient être transposés à ce cas et il serait évidemment intéressant de combiner espace de travail statique et espace de travail cinématique où coexisteraient des contraintes sur les forces et les vitesses articulaires. Dans ce cas, il serait nécessaire de réfléchir en profondeur sur l'implantation car la relative symétrie des matrices impliquées (J^{-1} et J^{-T}) pourrait être source de gain en temps de calcul comparé à deux algorithmes tournant en parallèle.

Il serait aussi intéressant de mieux comprendre la structure géométrique de W_S . Parfois deux composantes de l'espace de travail statique ne sont connectées que par une singularité. Peut-on alors partant d'une pose d'une composante rejoindre une pose d'une autre composante en passant par la singularité tout en respectant les contraintes sur les τ ? Quelle serait alors la stratégie de passage ?

- Dans ce document nous nous sommes concentré sur des robots avec un unique mode d'assemblage. Il serait donc intéressant d'étendre cette étude à des robots possédant plusieurs modes d'assemblage (par exemple passer du 3 – *RPR* à un robot de type 3 – *RRR*) en incorporant toutes les forces/moments des chaînes dans notre analyse. Dans ce cas, il serait important de distinguer plusieurs τ_{max} , l'un pour les composants passifs des chaînes, l'autre pour l'actionnement, ces deux éléments ayant des coûts différents. Dans le même ordre d'idée l'étude de robot à mobilité restreinte présenterait aussi un intérêt.

- L'espace de travail statique présente aussi un intérêt pour la planification de trajectoire, pour vérifier qu'une trajectoire, éventuellement incertaine (erreur de contrôle), pour un robot dont le modèle géométrique est affecté par des incertitudes est de manière garantie à l'intérieur de l'espace de travail statique et, si ce n'est pas le cas, proposer une stratégie afin de déterminer une trajectoire à l'intérieur de W_S . La notion d'espace de travail statique est également importante pour la conception, un sujet aussi très important pour l'équipe COPRIN. Il serait par exemple important de déterminer l'ensemble des paires (géométrie du robot, τ_{max}) conduisant à des espaces de travail statique qui incluraient un espace donné dans un cahier des charges. Une détermination par intervalles des paramètres de géométrie permettrait de prendre en compte les incertitudes de fabrication et donc de garantir que le robot réel satisferait au cahier des charges. La donnée simultanée du τ_{max} permet de plus de dimensionner mécaniquement les jambes

du robot (actionneur, articulations passives) ce qui a un impact non négligeable sur son coût. D'un point de vue de la conception, il est clair que l'aspect statique n'est qu'un élément parmi tant d'autres d'un cahier des charges. Toutefois notre approche a l'intérêt d'être intégrale dans une chaîne de module traitant séparément chaque aspect du cahier des charges. Ces modules prennent tous en entrée une spécification intervalles des paramètres de conception et en sortie une liste d'intervalles pour ces paramètres telle que la performance traitée par le module vérifie le cahier des charges. En enchaînant ces modules, on construit ainsi progressivement les robots qui vont satisfaire le cahier des charges ou, en cas d'échec, on identifie les performances du cahier des charges qui sont trop ambitieuses. L'intérêt de présenter un continuum de solution, outre la gestion des incertitudes est de permettre de prendre ultérieurement en compte des contraintes ne figurant pas dans le cahier des charges (aspects économique par exemple).

• Enfin du point de vue de l'analyse par intervalles il est clair que l'étude des continuum de solutions est très complexe et comme nous l'avons vu elle nécessite de définir plusieurs stratégies, plusieurs tests d'inclusion ou d'exclusion qui se révéleront plus ou moins efficace suivant la taille des boîtes et/ou la nature du problème. Aussi il peut-être intéressant de définir un outil de décision automatique permettant de choisir la bonne stratégie à adapter. Nous pensons que l'utilisation du calcul parallèle à ces fins peut se révéler très efficace. Bien que conscient de la difficulté d'une telle tâche, il nous semble cependant intéressant de regarder dans cette direction. D'un autre coté nous avons adapté une méthode de filtrage classique comme le rognage au cas des continuum de solutions. Il peut être intéressant de mesurer l'impact de cette méthode sur d'autres types de problèmes et espérer obtenir une amélioration conséquente.

BIBLIOGRAPHIE

- [1] M. Arsenault and R. Boudreau. The synthesis of three-degree-of-freedom planar parallel mechanisms with revolute joints (3-RRR) for an optimal singularity-free workspace. *J. of Robotic Systems*, 21(5) :259–274, 2004. (Cité page 21.)
- [2] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget. Revising Hull and Box Consistency. In *Proc. ICLP*, pages 230–244, 1999. (Cité page 54.)
- [3] S. Bhattacharya, H. Hatwal, and A. Ghosh. Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators. *Mechanism and Machine Theory*, 33(7) :965–974, October 1998. (Cité page 14.)
- [4] I. Bonev and C.M. Gosselin. Singularity loci of spherical parallel mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 2968–2973, Barcelona, April, 19–22, 2005. (Cité page 21.)
- [5] I.A. Bonev and D. Zlatanov. The mystery of the singular SNU translational parallel robot. www.parallemic.org/Reviews/Review004.html, June, 12, 2001. (Cité page 15.)
- [6] I.A. Bonev, D. Zlatanov, and C. Gosselin. Singularity analysis of 3 dof planar mechanisms via screw theory. *ASME J. of Mechanical Design*, 125(3) :573–581, September 2003. (Cité page 14.)
- [7] E. Borel. Mémoire sur les déplacements à trajectoire sphériques. *Mémoire présentés par divers savants*, 33(1) :1–128, 1908. (Cité page 14.)
- [8] R. Bricard. Mémoire sur la théorie de l’octaèdre articulé. *Journal de Mathématiques pures et appliquées, Liouville*, tome 3 :113–148, 1897. (Cité page 14.)
- [9] A. Cauchy. Deuxième mémoire sur les polygones et les polyèdres. *Journal de l’École Polytechnique*, pages 87–98, May 1813. (Cité page 14.)
- [10] I R. Clave. *Conception d’un robot parallèle rapide à 4 degrés de liberté*. PhD thesis, EPFL, Lausanne, 1991. n° 925. (Cité page 3.)
- [11] A. Dandurand. The rigidity of compound spatial grid. *Structural Topology* 10, pages 43–55, 1984. (Cité page 18.)
- [12] P. Dietmaier. The Stewart-Gough platform of general geometry can have 40 real postures. In *ARK*, pages 7–16, Strobl, June 29– July 4, 1998. (Cité page 12.)
- [13] C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *IEEE Trans. on Robotics and Automation*, 6(3) :281–290, June 1990. (Cité page 14.)

- [14] C. Gosselin, J. Sefrioui, and M.J. Richard. Solution polynomiale au problème de la cinématique directe des manipulateurs parallèles plans à 3 degrés de liberté. *Mechanism and Machine Theory*, 27(2) :107–119, March 1992. (Cité page 12.)
- [15] C. Gosselin and E. St-Pierre. Development and experimentation of a fast 3-dof orienting device. *Int. J. of Robotics Research*, 16(2) :619–630, 1980. (Cité page 3.)
- [16] V.E. Gough. Contribution to discussion of papers on research in automobile stability, control and tyre performance, 1956-1957. Proc. Auto Div. Inst. Mech. Eng. (Cité page 22.)
- [17] V.E. Gough and S.G Whitehall. Universal tire test machine. In *Proceedings 9th Int. Technical Congress F.I.S.I.T.A.*, volume 117, pages 117–135, London, May 1962. (Cité pages 3 et 22.)
- [18] M. Rueher H. Collaviza, F. Delobel. Comparing partial consistencies. *Reliable Computing*, 5(3) :213–228, 99. (Cité page 64.)
- [19] Eldon Hansen and G. William Walster. *Global Optimization using Interval Analysis, Second Edition*. CRC Press, 2003. (Cité page 64.)
- [20] Eldon R. Hansen. On solving systems of equations using interval arithmetic. *Mathematics of Computation*, 22(102) :374–384, 1968. (Cité page 50.)
- [21] Tsigaridas Hladik, Daney. An algorithm for the real interval eigenvalue problem. Technical report, INRIA Sophia-Antipolis, 2008. Research report 6680. (Cité page 73.)
- [22] M.L. Husty. An algorithm for solving the direct kinematic of Stewart-Gough-type platforms. *Mechanism and Machine Theory*, 31(4) :365–380, May 1996. (Cité page 22.)
- [23] M.L. Husty and A. Karger. Self-motions of Griffis-Duffy type parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 7–12, San Francisco, April, 24-28, 2000. (Cité page 6.)
- [24] C. Innocenti and V. Parenti-Castelli. Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators. In *22nd Biennial Mechanisms Conf.*, pages 553–560, Scottsdale, September, 13-16, 1992. (Cité page 21.)
- [25] Rohn J. Systems of interval linear equations and inequalities(rectangular case). Technical report, Institut of computer Science, Academy of Sciences of the Czech republic, 2002. Research report 875. (Cité page 64.)
- [26] A. Karger and M. Husty. Classification of all self-motion of the original Stewart-Gough platform. *Computer-aided design*, 30(3) :205–215, 1998. (Cité page 6.)
- [27] R.B Kearfott. Empirical evaluation of innovations in interval branch and bound algorithms for nonlinear algebraic systemes. *SIAM J. Sci Comput.*, 18(2) :574–594, 1997. (Cité page 66.)
- [28] Hunt K.H. *Kinematic geometry of mechanisms*. Clarendon Press, Oxford, 1978. (Cité page 14.)

- [29] X. Kong and C.M. Gosselin. Forward displacement analysis of third-class analytic 3-RPR planar parallel manipulators. *Mechanism and Machine Theory*, 36(9) :1009–1018, September 2001. (Cité page 22.)
- [30] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational complexity and feasibility of data processing and interval computations*. Kluwer, 1997. (Cité page 47.)
- [31] O. Lhomme. Consistency Techniques for Numeric CSPs. In *IJCAI*, pages 232–238, 1993. (Cité pages 54 et 64.)
- [32] D. Daney M. Gouttefarde, J.-P. Merlet. Wrench-feasible workspace of parallel cable-driven mechanisms. In *Proc. of the IEEE Int. Conf. Robotics and Automation (ICRA 07)*, pages 1492–1497, Rome, Italie, 2007. (Cité page 64.)
- [33] B. Mayer St-Onge and C.M. Gosselin. Singularity analysis and representation of the general Gough-Stewart platform. *Int. J. of Robotics Research*, 19(3) :271–288, March 2000. (Cité page 21.)
- [34] J-P. Merlet. ALIAS : An Algorithms Library for Interval Analysis for Equation Systems. Technical report, INRIA Sophia, 2000. www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS.html. (Cité page 54.)
- [35] J-P. Merlet. *Parallel Robots*. Springer, 2006. (Cité pages 3 et 17.)
- [36] J-P. Merlet. Jacobian, manipulability, condition number and accuracy of parallel robots. In *ISRR*, San Francisco, October, 12-15, 2005. (Cité page 21.)
- [37] J-P. Merlet. Singular configurations of parallel manipulators and Grassmann geometry. *Int. J. of Robotics Research*, 8(5) :45–56, October 1989. (Cité pages 16 et 18.)
- [38] J-P. Merlet. Jacobian, manipulability, condition number, and accuracy of parallel robots. *ASME J. of Mechanical Design*, 128(1) :199–206, January 2006. (Cité page 21.)
- [39] J-P. Merlet. A formal-numerical approach for robust in-workspace singularity detection. *IEEE Trans. on Robotics*, 23(3) :393–402, June 2007. (Cité page 21.)
- [40] A.P. Murray and F. Pierrot. N-position synthesis of parallel planar RPR platforms. In *ARK*, pages 69–78, Strobl, June 29- July 4, 1998. (Cité page 22.)
- [41] Revol N. Introduction à l’arithmétique par intervalles. Technical report, INRIA Sophia, 2001. (Cité page 49.)
- [42] D.N. Nenchev and M. Uchiyama. Singularity-consistent path planning and control of parallel robot motion through instantaneous-self-motion type. In *IEEE Int. Conf. on Robotics and Automation*, pages 1864–1870, Minneapolis, April, 24-26, 1996. (Cité pages 14 et 21.)
- [43] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990. (Cité pages 57 et 58.)

- [44] Veblen O. and Young J.W. *Projective Geometry*. The athenaeum press, 1910. (Cité page 15.)
- [45] J.F. O'Brien and J.T. Wen. Kinematic control of parallel robots in the presence of unstable singularities. In *IEEE Int. Conf. on Robotics and Automation*, pages 3154–3159, Seoul, May, 23-25, 2001. (Cité pages 14 et 21.)
- [46] M.K. Park and J.W. Kim. Kinematic manipulability of closed chains. In *ARK*, pages 99–108, Portoroz-Bernadin, June, 22-26, 1996. (Cité page 15.)
- [47] H. Pottmann, M. Peternell, and B. Ravani. Approximation in line space. Applications in robot kinematics. In *ARK*, pages 403–412, Strobl, June 29- July 4, 1998. (Cité page 18.)
- [48] Siegfried M. Rump. Algorithms for verified inclusions—theory and practice. pages 109–126, 1988. (Cité page 41.)
- [49] S. Sen, B. Dasgupta, and A.K. Mallik. Variational approach for singularity-path planning of parallel manipulators. *Mechanism and Machine Theory*, 38(11) :1165–1183, November 2003. (Cité pages 14 et 21.)
- [50] D. Stewart. A platform with 6 degrees of freedom. *Proc. of the Institution of mechanical engineers*, 180(Part 1, 15) :371–386, 1965. (Cité page 3.)
- [51] L-W. Tsai. Kinematics of a three-dof platform with three extensible limbs. In *ARK*, pages 401–410, Portoroz-Bernadin, June, 22-26, 1996. (Cité page 13.)
- [52] P.A. Voglewede and I. Ebert-Uphoff. Measuring "closeness" to singularities for parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 4539–4544, New Orleans, April, 28-30, 2004. (Cité page 21.)
- [53] M. Zein, P. Wenger, and D. Chablat. Singular curves in the joint space and cusp points of 3-RPR parallel manipulators. *Robotica*, 25(6) :712–724, November 2007. (Cité page 22.)
- [54] D. Zlatanov, I.A. Bonev, and C.M. Gosselin. Constraint singularities of parallel mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 496–502, Washington, May, 11-15, 2002. (Cité page 15.)
- [55] D. Zlatanov, R.G. Fenton, and B. Benhabib. Identification and classification of the singular configurations of mechanisms. *Mechanism and Machine Theory*, 33(6) :743–760, August 1998. (Cité page 15.)